

AN EXPERT SYSTEM FOR AUTOMATED PROCESS PLANNING
OF TURNED COMPONENTS

by

SRIDHAR KOTHANDARAMAN

B.E. Bharathiar University, May 1986

A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Approved by:



Major Professor

TABLE OF CONTENTS

Title Page	i
Table of Contents	ii
List of Figures	v
List of Tables	vi
Acknowledgements	vii
1.0 INTRODUCTION	1
1.1 EXPERT SYSTEMS	2
1.1.1 Definitions	2
1.1.2 Characteristics of Expert Systems	3
1.1.3 Applications	3
1.2 DESIGN AND MANUFACTURING	4
1.3 LITERATURE SURVEY	7
1.4 THESIS OBJECTIVES	9
2.0 DATA REPRESENTATION	12
2.1 FEATURE-BASED MODELS	14
2.1.1 Definition of a Feature	14
2.1.2 Design and Manufacturing Features	15
2.1.3 Requirements of a Feature-based Model ..	16
2.1.4 Advantages of a Feature-based Model	16
2.2 DEVELOPMENT OF A DATA STRUCTURE FOR FEATURE- BASED DESCRIPTION OF TURNED COMPONENTS	17
2.2.1 Feature Data	18
2.2.2 Environmental Data	24
3.0 SYSTEM KNOWLEDGE	26

3.1	TYPES OF KNOWLEDGE	27
3.2	OBTAINING INPUT INFORMATION	29
3.3	SELECTION OF MACHINE AND WORK-HOLDING DEVICE ..	30
3.4	SELECTION OF A HOLDING SURFACE	32
3.5	SELECTION OF OPERATION	34
3.6	CALCULATION OF SPEED	36
3.7	MODIFICATION OF CURRENT PART DESCRIPTION	37
3.8	CALCULATION OF FEED RATE	40
3.9	OUTPUT	41
3.10	SEMANTIC MODELS	41
3.10.1	Semantics for Part Modification	43
3.10.2	Semantics for Holding Surface Selection	52
4.0	KNOWLEDGE-BASE IMPLEMENTATION	56
4.1	PC PLUS - AN OVERVIEW	56
4.1.1	Features of PC Plus	57
4.1.2	Frames	58
4.1.3	Parameters	60
4.1.4	Variables.....	61
4.1.5	Rules	62
4.1.6	Functions	62
4.2	STRUCTURE OF OUR KNOWLEDGE BASE	63
4.2.1	Frame PROCESSOR	66
4.2.2	Frame INPUT	66
4.2.3	Frame INPUT-LIST	67
4.2.4	Frame HOLDING	67

4.2.5	Frame CHECKSTOCK	68
4.2.6	Frame BAR-STOCK	68
4.2.7	Frame MODBAR	69
4.2.8	Frame CAST-STOCK	69
4.2.9	Frame HOLD-SURFACE	70
4.2.10	Frame OPSEQUENCE	70
4.2.11	Frame MODIFY	71
4.2.12	Frame SURFACE-FINISH	71
4.2.13	Frame PRINT-OUTPUT	72
5.0	RESULTS	73
6.0	CONCLUSIONS AND SUGGESTIONS	110
6.1	CONCLUSIONS	110
6.2	SUGGESTIONS	113
	REFERENCES	115
	APPENDIX A DATABASE OF SPEEDS AND FEEDS	117
	APPENDIX B TABLE FOR FINDING CUTTING FORCE CONSTANT	123

LIST OF FIGURES

Figure 2.1	Example of a Part Description	22
Figure 3.1	Example of Part Modification	38
Figure 4.1	Structure of our Knowledge Base	64
Figure 5.1	Operation Sequence for Example 1	77
Figure 5.2a	Operation Sequence for Example 2	80
Figure 5.2b	Operation Sequence for Example 2	81
Figure 5.3a	Operation Sequence for Example 3	84
Figure 5.3b	Operation Sequence for Example 3	85
Figure 5.4a	Operation Sequence for Example 4	88
Figure 5.4b	Operation Sequence for Example 4	89
Figure 5.4c	Operation Sequence for Example 4	90
Figure 5.5a	Operation Sequence for Example 5	93
Figure 5.5b	Operation Sequence for Example 5	94
Figure 5.6a	Operation Sequence for Example 6	97
Figure 5.6b	Operation Sequence for Example 6	98
Figure 5.7a	Operation Sequence for Example 7 (Part 1) ...	101
Figure 5.7b	Operation Sequence for Example 7 (Part 2) ...	104
Figure 5.8a	Operation Sequence for Example 8	107
Figure 5.8b	Operation Sequence for Example 8	108
Figure 5.8c	Operation Sequence for Example 8	109

LIST OF TABLES

Table 2.1	Design Feature Descriptors and Attributes ...	21
Table 2.2	Manufacturing Feature Descriptors and Attributes	23
Table 5.1	Design Feature Descriptors and Attributes ...	74
Table 5.2	Manufacturing Feature Descriptors and Attributes	74

ACKNOWLEDGEMENTS

I would like to thank

Dr. Prakash Krishnaswami, my mentor, advisor and friend without whose untiring patience and suggestions the completion of this thesis would have been impossible.

Dr. L. T. Fan for allowing me access to equipment in the Department of Chemical Engineering and accepting to be my committee member.

Dr. Garth Thompson for serving on my committee.

The Centre for Research in Computer-controlled Automation at Kansas State University for providing the facilities and equipment used in this project.

The National Science Foundation for providing support through grant number DMC-8705212.

The Department of Mechanical Engineering for all the support and equipment provided.

My friend Ajit Devotta for sharing his manufacturing knowledge with me and for his patience in handling my oft-repeated questions.

My friends for putting up with me.

CHAPTER 1

INTRODUCTION

In today's world, most people in the technical workplace are oriented towards a well-defined area of specialization. The process of transferring the expertise of these people to newer employees is generally inefficient and often results in a loss of valuable information. The acquisition of the expertise of these people and its formulation in a manner suitable for use by inexperienced users is, therefore, of great importance to industry. A common need of a large number of companies is for experts to solve machinery or process problems. However, a large company might have only two to five experts to cover as many as 100 different machines in perhaps twenty five plants [5]. Although these experts could keep up with problem solving, they have little time left for development activities or process improvement. In most of these companies, diagnostic or Artificial Intelligence (AI) systems could be successfully implemented.

In AI, the representation of knowledge is a combination of structuring information and formulating procedures for operating on this information. The basic components of knowledge can be identified as :

- i) Facts
- ii) Rules
- iii) Heuristics

Facts detail the pertinent information about a given situation or environment. How to use the facts is described in the statement of rules. Any information that is useful in solving a particular problem that is not explicitly stated falls into the category of heuristics. Typically, facts and rules are widely available. Heuristics on the other hand, are gained only through experience and training [7].

1.1 EXPERT SYSTEMS

Expert systems are a common application of artificial intelligence in manufacturing, electronics, medicine, chemistry, geology and several other areas. Expert systems are tools by which abstract information can be gathered and pieced together to accomplish a specific set of tasks that are normally performed by a human expert. Expert systems are gaining popularity in a variety of fields where their use simplifies the complicated tasks that must be performed. In the following sections, we discuss the definitions, features and areas of applications of expert systems.

1.1.1 Definitions

It is difficult to arrive at a precise definition of an expert system. It can be loosely described as a software package that captures and manipulates the knowledge and strategies that a human expert applies to solve a problem. An expert system is made up of a knowledge base and an

inference engine. The knowledge base consists of facts, rules about relationships among facts, strategies for making educated guesses and rules of thumb developed from experience. An inference engine is the mechanism that performs the inference procedure to arrive at a conclusion. The inference engine makes use of the knowledge stored in the knowledge base to make a decision.

1.1.2 Characteristics of Expert Systems

Expert systems share some common characteristics with traditional software programs. They both reach conclusions based on what they are programmed to do and both communicate with the client or user through a suitable user interface. In addition, expert systems have the following distinguishing characteristics:

- * They handle uncertain data or incomplete information.
- * They justify their conclusions (explain how conclusions were reached).
- * They provide convenient facilities for adding new conditions or rules with usage.
- * They provide relatively easy handling of mixed numeric and non-numeric data.

1.1.3 Applications

Expert systems are capable of handling uncertain data or information and are therefore suitable not only for applications involving analysis of data, but also for appli-

cations where the decision process or input information includes a degree of uncertainty. Since expert systems are built to handle conditions and rules, they can be used effectively in problems where the domain knowledge can be structured as a set of conditional relationships.

The capabilities of expert systems can be utilized for the extensive decision making required in manufacturing. The communication gap between design and manufacturing is a critical area where expert systems could play a vital role. The following sections discuss Computer-Aided Design (CAD), Computer-Aided Manufacture (CAM) and Computer-Aided Process Planning (CAPP).

1.2 DESIGN AND MANUFACTURING

Conventional mechanical design can be greatly expedited through the effective use of computers. Computer-aided design is a process whereby one can generate, evaluate and modify the model of the object one wants to produce through interactions with a computer. The process lasts until a model which satisfies the given criteria on performances, reliability, producibility, maintainability, cost, etc. is obtained. Computers programs for design simplify the design process to a large extent. Most computer-aided design programs are written to be interactive and user friendly. Computer-aided design systems are characterized by the large

volume of data handled and the large number of calculations that must be performed.

The development of computer-aided manufacture is closely associated with the growth in the popularity of Numerical Control (NC) machine tools. On these machines, the manufacturing process is controlled by NC code residing on a magnetic tape or in computer memory. The function of the human operator is limited to tasks like loading and unloading components, loading and maintaining tools in proper condition, overall maintenance and programming.

Computer-aided manufacturing has numerous advantages compared to conventional manufacturing. Higher accuracy in machining resulting in finer tolerances are easily obtained, along with greater repeatability. Increased productivity and lower scrap levels are also realizable through computer-aided manufacturing. Computer-aided manufacturing at its current state of development deals mostly with Computerized Numerical Control (CNC), Direct Numerical Control (DNC) and NC machine tools. Other types of machines such as single-spindle automats and conventional lathes are almost completely ignored.

The output of the computer-aided design process is often unsuitable as input to a computer-aided manufacturing system, which requires other manufacturing data related to

the designed component. The data required for manufacturing the product is obtained by generating a process plan for the component.

Once the design of a product is finalized, the process plan for manufacturing the product is constructed. The process plan forms a vital link between design and manufacture. The process plan includes selection of machines and work-holding devices, operation sequences, tolerancing, routing, fixturing, tooling, operating instructions, methods, time standards and required drawings. The process planning starts with the selection of a machine for the part to be machined. As soon as the machine is selected a work-holding device is to be selected from the available devices for that particular machine. The operations sequence is then generated for that work-holding device. The holding surface needs to be found in order to determine which surfaces are machinable with that holding. The calculation of the holding surface depends on the tolerance requirements on the part. The process plan generation is quite complicated and requires a very high skill level.

The generation of a correct process plan is critical since the efficiency of manufacture of the product is determined by the process plan. For example, for a rotational component, if a process plan with several reversals of the part is generated where another with fewer reversals is

possible, the efficiency is reduced in manufacturing and the cost increases.

1.3 LITERATURE SURVEY

Expert systems for operations planning are limited in number, since the field is relatively new. There is a lot of current research being done in this field but there are very few published papers in the literature. This section discusses the recent research in this area and briefly describes some of the computer-aided process planning systems that have been developed.

Spur, Krause, Mayr, Vassiliadis and Siebmann [14] developed a system for processing geometric data. This system, called "Baustein GEOMETRIE", handles interactive elemental inputs like points, lines, circles, rectangular prisms, cylinders and cones. It should be noted here that a geometrical model is unsuitable for incorporation into a knowledge base due to the voluminous numerical data that is used to describe the elements.

General Electric Company has developed a computer-aided process planning system for aircraft engine rotating parts [16]. This system is an overall process planning system and detailed operations are left to the user to be entered interactively. The selection of speeds, feeds and tool changes are accomplished by economic analysis programs and

the process planner is allowed to choose the machining conditions based on the cost and production rate.

The AUTOTECH program system [15] is a computer-aided process planning system developed in West Germany. This system, like most other CAPP systems, does not perform operations planning. AUTOTECH does computer-aided calculations of the operational or machining parameters and machining times. The system does not provide for any intelligent decision making capability.

Barkocy and Zdeblick [1] have developed a knowledge based system for operations planning, called CUTTECH. This system makes use of metal-cutting data and technology to recommend tools and cutting parameters for machining. Cutting tools, cut sequences, speeds and feeds are selected for a user-defined part feature. The operation sequence is arrived at based on the assumption that only NC machines are used for the manufacture of the component. The system does not take into account the fact that the operation sequence is affected by the machine used and the work-holding device selected.

Engineers at Bendix Corporation [4] are developing an expert system for generative process planning using a feature-based input. The features are extracted interactively from a solid modeler and fed into the rule-based expert

system which performs the process planning.

A feature-based object-oriented CAD/CAM environment tailored for turned components was developed by Unruh [17]. This environment provides numerous design and manufacturing capabilities such as NC code generation and interactive modeling.

1.4 THESIS OBJECTIVES

As already noted, knowledge-based systems for operations planning are limited in number. Existing CAPP systems perform an overall process planning with very little emphasis on detailed operations planning. In almost all cases, critical decisions are left to the user and must be input interactively. At most, these systems perform a generalized selection of machines through which the part or component has to be processed, and a small amount of operations planning on each of the machines. Importance is given to routing the component through different machines rather than individual machining sequences on a single machine. Even in the limited operations planning being done, all factors involved are not taken into account. Factors like tolerance and surface finish are not taken into consideration while calculating speeds and feeds for individual operations. Also, no inherent knowledge or machining expertise is provided in the existing systems.

The objective of the present work was to develop a knowledge-based system that would overcome most of the above-mentioned drawbacks. In developing this system, turned components that could be machined on an NC machine or a single spindle automat were considered. Efforts were also made to account for the design tolerances and surface finishes at the operations planning stage to determine the manufacturability of the component. The various stages of development of the system are listed below :

1. Formulation and standardization of the data structures needed for the internal representation of components and manufacturing operations.
2. Development of the rules required for data manipulation.
3. Formulation and implementation of the rules required for the selection of a machine and work-holding device; implementation of an analysis module for the calculation of machining tolerances and selection of holding surface for the given part geometry.
4. Implementation of expert knowledge for deciding the machining sequence as well as cutting speeds and feeds for each operation.

In chapter 2, the relative merits of geometry-based descriptions and feature based descriptions of mechanical components and assemblies is discussed. The data structures

used in the knowledge base and the machining parameters database are also described in this chapter. The basis on which machining sequences are determined is discussed in chapter 3. The effect of parameters like tolerance and tool material on machining sequence selection and the effects of work-holding and machine selected are discussed. A set of rules for determining the machining sequence is also presented. In addition, a formal semantic definition of the necessary data modification operations is derived. In chapter 4, implementation of the system as a knowledge base is discussed. An overview of the expert system shell and the structure of the knowledge base are given. Each frame and its function in the knowledge are discussed in detail. Chapter 5 discusses the results obtained from several examples run on the developed system. Examples with external operations and internal operations were run. Different options were selected for each example and the results obtained were reasonable. Chapter 6 summarizes the research completed and suggests areas for future research and investigation.

CHAPTER 2

DATA REPRESENTATION

Full integration and automation of design and manufacture is realizable only if a complete description of the component of interest is generated during the design phase. It is not possible to achieve high efficiency in a CAD system unless it has the capability to store, manipulate and use the knowledge required for intelligent interpretation of the designer's commands [12].

The geometric definition of a model is the baseline for the design within a CAD/CAM environment. Other non-geometric data are normally needed to complete the definition of a model at the design stage.

Geometric modeling, the complete representation of the shape of components, is generally accepted as a prerequisite for the development of comprehensive computer-aided design and production systems [6]. This form of data representation for computerized process planning is generally accomplished by making use of a geometry-based interactive CAD system to obtain a description of the part. Most CAD/CAM and CAPP systems in existence today use geometric representation for describing a component. In these systems, a component is typically described in terms of its geometric primitives: lines, circles, polygons, cylinders, cones,

prisms, spheres and so forth.

In a geometry-based descriptive system, the most frequent functions to be performed are [2]:

- * Definition of basic geometric elements: points, straight lines, circles, conics, planes, cylinders, cuboids, etc. The definition may be done explicitly or implicitly through relational constraints like connectivity, parallelism, and tangency.
- * Combination of these elements in order to define complex surfaces and volumes.
- * Geometric transformations: translations, rotations, scaling, etc.
- * Modification of surfaces and volumes to reflect manufacturing operations such as drilling, milling, planing.

In such a system, data is most commonly stored in some standardized parametric form. For example, we may store the x and y coordinates of the endpoints for a straight line; x and y coordinates of the center and the radius or diameter for a circle; and so on. Data of this form is limited to representing the geometric aspects of a component. The associated manufacturing and design data are not represented by a geometry-based system. Additional information like tolerance, surface finish, etc. is required for manufacturing the part. This information can not be conveyed easily

from design to manufacture in a geometry-based system. Furthermore, designers generally conceptualize in terms of functional entities like steps, tapers, threads, holes and so forth. Geometry-based models, on the other hand, are expressed in terms of geometric entities like lines and circles. Thus, a geometry-based CAD system does not provide a natural environment for the designer. These shortcomings provide the motivation for developing a feature-based representation of the class of components to be considered.

2.1 FEATURE-BASED MODELS

2.1.1 Definition of a Feature

There is no universally accepted definition of a feature. Luby [9] defines a feature as a geometric form or entity whose presence or dimensions are relevant to one or more CIM functions or whose availability to a designer as a primitive facilitates the design process. For process planning systems, entities like steps, tapers, holes, etc. can serve as features representing the part to be machined. For each feature to be completely defined, several associated properties must be added as part of the data defining the feature. These properties are often called the attributes of the feature. These attributes comprise both geometric and non-geometric information. For example, non-geometric information such as tolerance and surface finish could be included as attributes of features in describing a rotation-

al component.

2.1.2 Design and Manufacturing Features

Features can be classified into design and manufacturing features. In both cases, the general idea is to define a set of primitive features and use combinations of these features for describing the design/manufacture of a component. However, the nature of the features used in each case is quite different. The primitive design features generally represent actual solids and the operations by which features are combined are usually constructive in nature. In manufacture, on the other hand, the volumes of material removed by machining processes serve as the primitive features. Manufacturing features represent the volume and shape of the material to be removed from the raw part to produce the finished part. It is critical to distinguish between design features and manufacturing features. Design features effectively describe a component to be manufactured. Manufacturing features describe the operations that must be performed on an unfinished component to produce the component specified by the designer. Manufacturing features are therefore the means by which a designed component is to be produced.

The above differences between design and manufacturing features creates a compatibility problem in that different

sets of features have to be dealt with during integration of design and manufacture. We must either devise an unified scheme for representing both design and manufacturing features or relate them to each other. In the proposed system a single format is used for representing the features for design and manufacture. Even with the same format, the attributes of the design features are modified to obtain manufacturing features. This similarity in structure facilitates easy handling capabilities.

2.1.3 Requirements of a Feature-based Model

Any feature-based representation should be a complete description of geometric and technical information. It should also be structured in such a way that a relatively inexperienced user can easily understand and work with the feature-based description. All operations on a feature or an ensemble of features should be based on a formal semantic model to ensure consistency and correctness. Finally, the representation should provide for the integration of design and manufacture by supporting design and manufacturing operations on the same data model.

2.1.4 Advantages of a Feature-based Model

Feature-based models have the following advantages when compared to geometry-based models:

1. Specific characteristics of the feature being described are accounted for along with the required

geometric data.

2. A feature-based model gives a better understanding of the shape of the component compared to a geometry-based model. For example, a THREAD (a feature) could be easily visualized compared to a HELIX of finite width and depth inscribed on a cylinder (a geometry-based descriptor).
3. Feature-based models are more flexible. Multiple features can be combined to form a complex feature and more attributes can be added to an existing feature. Addition of new features at a later stage of development is done with ease in a feature-based system.
4. Designers think in terms of features rather than geometric entities like lines or circles. Consequently, a feature-based representation simplifies the design process to a great extent.

2.2 DEVELOPMENT OF A DATA STRUCTURE FOR FEATURE-BASED DESCRIPTION OF TURNED COMPONENTS

The development of a feature-based model conforming to the above requirements involves several stages :

1. Identification and grouping of essential features in the part.
2. Gathering required geometric data for each distinctive feature.

3. Addition of design and manufacturing data for that feature, and
4. Selection of proper data structure to suit specific requirements like ease of handling and data manipulation.

The common link running through the whole design / manufacturing process is data [6]. From the initial design to the finished product, a vast amount of information is accumulated and processed. The key to the development of computer-aided process planning systems is the structuring of all data concerning geometry, fabrication, facilities, tooling and materials into categories and logical relationships.

In our proposed system, two forms of data have been identified. They are:

1. Feature data and
2. Environmental data.

In the following sections, detailed descriptions of both types of data are given.

2.2.1 Feature Data

As noted earlier, features are generally classified into design and manufacturing features. Any process planning system requires a raw-part description and a finished-part description in order to determine an operation se-

quence. It is clear that the raw-part description and finished-part description are representative of material present in the component and hence should be modeled using design features. On the other hand, each operation in the operation sequence represents the removal of material from the component and should therefore be modeled using manufacturing features.

For rotational components, the basic design features required can be identified as STEPS, TAPERS, THREADS, FILLETS, GROOVES, CHAMFERS, HOLES, TAPERED HOLES and THREADED HOLES. Each of the features selected above requires certain basic geometric data that would specify its the location and size. For a STEP, the starting and ending x-coordinates and a diameter are essential. The starting and ending x-coordinates and diameters are required to be specified in the case of a TAPER. A THREAD requires the starting and ending x-coordinates, diameter, pitch and thread profile to describe it completely. FILLETS and GROOVES need a starting x-coordinate and a radius to be identified. A starting x-coordinate and an angle are required to specify a CHAMFER. A HOLE has the same geometric data as for a STEP, while a TAPERED HOLE has a specification that is identical to a TAPER. A THREADED HOLE has a description that is similar in structure to a THREAD.

Along with the geometric data, each feature requires design and manufacturing data which facilitates the generation of a process plan. The most critical design data for rotational components are the design tolerance and surface finish. These are included in the attributes of each feature along with the geometric data. This feature definition scheme is similar to the one developed by Unruh [17].

Having defined the features and their attributes, the next task is to select a suitable data structure. The requirements for our data structure can be generalized as follows:

1. Modifications to the part should be possible with minimum effort.
2. Efficient storage is essential.
3. Capability to handle both numeric and non-numeric data is required.

All of the above requirements can be realized with a list based data structure. Lists can be easily manipulated using LISP functions supported by most expert system shells. Lists make it possible to store information about the whole component in a single location. Modifications are accomplished by simple LISP operations. Finally, lists have the capability of maintaining both numeric and non-numeric data. Since turned components are composed of a linear sequence of features, a single list which is composed of the individual

feature description lists of the constituent features could be used for representing the entire part.

The maximum number of fields required by a feature is eight (for a thread), and the minimum is four (for a fillet, groove or chamfer). In order to simplify the structure of the list, a field length of eight was selected for feature description. This uniformity is preferred because the coding is greatly simplified and streamlined. The list representation of each design feature is summarized in Table 2.1. The tolerances, coordinates, radii, diameters, and pitches must be specified in inches, surface finishes in microinches , angles in degrees and the form is either square-thread or v-thread.

Feature Class	Attributes							
	1	2	3	4	5	6	7	8
Step	STEP	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dummy	Dummy
Taper	TAPER	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Thread	THREAD	Tol.	Sur.Fin.	X-Start	X-End	Dia.	Pitch	Form
Fillet	FILLET	Tol.	Sur.Fin.	X-Start	Radius	Dummy	Dummy	Dummy
Groove	GROOVE	Tol.	Sur.Fin.	X-Start	Radius	Dummy	Dummy	Dummy
Chamfer	CHAMFER	Tol.	Sur.Fin.	X-Start	Angle	Dummy	Dummy	Dummy
Hole	HOLE	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Taper	ITAPER	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Thrd.	ITHREAD	Tol.	Sur.Fin.	X-Start	X-End	Dia.	Pitch	Form

Table 2.1 Design Feature Descriptors and Attributes.

The raw-part and finished-part descriptions are composed of a sequence of features. The raw-part description is maintained in a single list, whereas the finished-part

description is maintained in two lists; one for external features and another for internal features. In all the three cases, the part description starts from the left end and proceeds in sequential order to the other end. That is, the leftmost feature has an x-coordinate of zero. The next feature to the right comes next in the list and so on. FILLETS, GROOVES and CHAMFERS are specified with their starting x-coordinate overlapping the feature on which they occur.

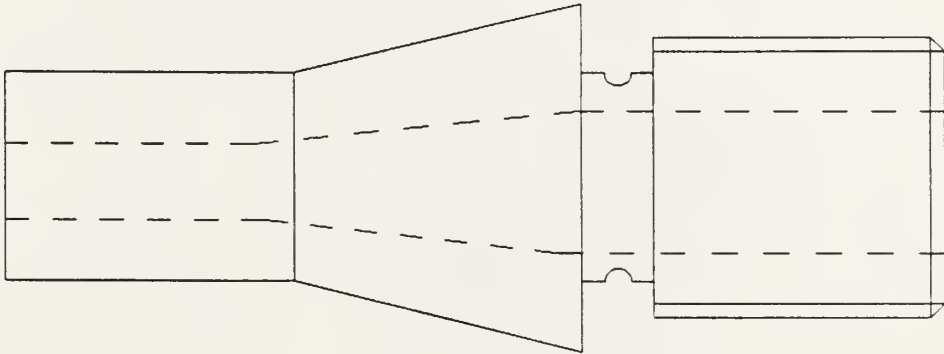


Figure 2.1 Example for Part Description.

The example shown in Figure 2.1 has the following descriptions:

External:

```
(STEP 0.01 20. 0. 2. 1.5 1.5 0. TAPER 0.01 20. 2. 4.
1.5 2.5 0. STEP 0.01 15 4. 4.5 1.5 1.5 0. GROOVE 0.01
30. 4.25 .3 0. 0. 0. THREAD 0.001 25. 4.5 6.5 2. .8
V-THREAD)
```

Internal:

(HOLE 0.002 25.0 0. 1.75 .75 .75 0. ITAPER 0.02 30.0
1.75 4. .75 1.0 0. HOLE 0.03 30. 4. 6.5 1.0 1.0 0.)

The operation description makes use of manufacturing features. The specification of the manufacturing feature should include sufficient geometric information to represent the geometry of the component after the completion of the manufacturing operation. In the present case, the geometric component of a manufacturing feature was defined to be the

Feature Class	Attributes							
	1	2	3	4	5	6	7	8
Step	STEP	Speed	Feed	X-Start	X-End	Dia.1	Dummy	Dummy
Taper	TAPER	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Thread	THREAD	Speed	Feed	X-Start	X-End	Dia.	Pitch	Form
Fillet	FILLET	Speed	Feed	X-Start	Radius	Dummy	Dummy	Dummy
Groove	GROOVE	Speed	Feed	X-Start	Radius	Dummy	Dummy	Dummy
Chamfer	CHAMFER	Speed	Feed	X-Start	Angle	Dummy	Dummy	Dummy
Hole	HOLE	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Taper	ITAPER	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Thrd.	ITHREAD	Speed	Feed	X-Start	X-End	Dia.	Pitch	Form

Table 2.2 Manufacturing Feature Descriptors and Attributes. geometry of the design feature that is created as a result of the operation that the manufacturing feature represents. Consequently, the manufacturing features are similar to the design features used for part description. The tolerance field in the part description is replaced by the cutting speed and the surface finish field is replaced by the feed rate of the tool. This takes care of the manufacturing data required for machining. The list representation of each

manufacturing feature is given in Table 2.2. The speed is in RPM, the feed rate is in inches per revolution, the coordinates, diameters, radii and pitches are in inches, the angles are in degrees and the form is either square-thread or v-thread.

The operation sequence is maintained in a list which contains information about the holding surface, operation description and reversals of part. The holding surface is specified by the starting and ending coordinates of the surface to be held. The operation description is a list of manufacturing features. As each operation is decided, the operation sequence list expands to accommodate the manufacturing feature representing that operation. Each new operation description is appended to the end of the existing operation sequence list. An example of an operation sequence list is given below:

```
(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2. FACE
0.25 STEP 204.65 0.025 2. 6.75 2.5 2.5 0 STEP 229.2
0.022 4. 4.75 2. 2. 0. REVERSE OBJECT HOLD THE OBJECT
BEGINNING AT 4.75 AND ENDING AT 6.75 STEP 220.38 0.0204
0. 2. 1.5 1.5 0)
```

2.2.2. Environmental Data

The features described in the previous section have associated attributes that are specific to them. In order

to perform the process planning of the component, additional information common to the entire part is required. This information forms the environmental data. The following environmental data are required for the system:

- * The types of machines available
- * The work-holding devices available
- * The stock material (Cast, Semi-finished or Bar)
- * The tool material
- * The tool breadth
- * The tool width
- * The tool overhang
- * The tool lip-angle
- * The work-piece SAE number
- * The machine-tool error.

If the component consists of features made of different materials, then it is necessary to include the stock material as a feature attribute. In our system, since we are considering turned components made of a single material, the stock material is considered to be part of the environmental data rather than feature data.

In the previous sections we have developed a representation scheme for a feature-based description of turned components. An implementation of a knowledge-based process planning system based on this scheme is discussed in detail in the next two chapters.

CHAPTER 3

SYSTEM KNOWLEDGE

A knowledge-based system requires facts and logical relationships to accomplish a goal or a set of goals. In this chapter, we will be discussing the various factors that affect the goals set and the manner in which knowledge is imparted to the system.

First, let us define the goals that are to be achieved by our system :

1. Obtain input information that defines the problem to the system.
2. Selection of a machine to be used for machining the part and a work holding device for clamping the component.
3. Selection of a holding surface on the component and determination of whether reversal is necessary.
4. Selection of the operation to be done on the part.
5. Calculation of the speed of rotation of the work-piece for that particular operation.
6. Modification of the component description to reflect the operation that has been completed.
7. Verification of current component description against desired finished component description. If verification fails, goals 3 through 6 above must be satisfied again.

8. Calculation of feeds for all the operations on the part.
9. Output the results of the consultation in a format suitable for human reading.

The above tasks have to be done sequentially. As soon as the required input information has been obtained, the machine and workholding device to be used for machining are selected. Then, if the material is cast or semi-finished stock, a suitable holding surface is determined. If the material is a bar stock, then holding surface determination is not necessary. The operation to be performed is selected and the cutting speed for the operation is calculated. All possible operations with this holding are performed. If the stock is cast or semi-finished and some more operations are required to be done, then the work-piece is reversed and a new holding surface is determined. The selection of operations is then continued. After all the operations are selected, the feed rate of the tool for each operation is calculated and the results are presented to the user. In the following sections, we will be discussing the above tasks in detail.

3.1 TYPES OF KNOWLEDGE

Knowledge is imparted to the system in several forms. Three kinds of knowledge can be distinguished in our system. They are

1. Rule-based knowledge
2. Formula-based knowledge and
3. Data-based or table-based knowledge.

Rule-based knowledge is generally in the form of IF-THEN clauses. Actions in the THEN clause are executed only if the conditions in the IF clause are met. Rule-based knowledge is generally used for backward chaining or forward chaining through the knowledge-base system. Backward chaining is the process whereby the system starts from a set goal and determines all the values required to achieve the goal in reverse sequence. Forward chaining is the process whereby the system follows a predetermined path to obtain all the values required to achieve the goal in forward sequence. A major problem in a rule-based knowledge representation is conflict resolution. During a consultation, a situation may occur where the if clauses of several are satisfied. The system then has to decide which of these rules should be fired. In designing an expert system, there are two ways to handle this problem. The first approach is to allow such conflicts to arise and include strategies in the knowledge base for handling these conflicts whenever they occur. The second approach is to structure the knowledge base in such a way that a conflict never arises i.e., the knowledge base design ensures that the if clause of only one rule will be satisfied at any point in the consultation. The latter

approach has been used our knowledge base.

Formula-based knowledge is a special case of rule-based knowledge. In the case of formula-based knowledge, values are assigned to parameters or variables based on a mathematical relationship to other parameters and variables.

Data-based or table-based knowledge is characterized by the determination of values for parameters or variables from a preset database, such as a look-up table. Conditional checks are performed to assign correct values from the database to the parameters.

3.2 OBTAINING INPUT INFORMATION

Defining the problem and the environment to the system is the first step in the solution process. The user is required to input the type of machines available for machining the component. Two types of machines provided are for: Numerically Controlled Turning Center and Single Spindle Automat. The user is given the option of selecting either one or both of the machines.

The two types of work-holding devices available for the user to choose are a chuck and a collet. Again, the user is left to choose either one or both of the devices.

The user is also prompted for :

The stock material (bar, cast or semi-finished)

The tool material (High Speed Steel (HSS), Cast

alloy (CA), Cobalt HSS (CHSS) or Sintered Carbide
(SC)}

The tool breadth

The tool width

The tool overhang

The tool lip-angle (ranging from 55 to 90 degrees)

The work-piece SAE number

The machine tool error

The work-piece SAE numbers are retrieved from a database and presented to the user for selection. Input information for the external finished component description, the internal finished component description and the raw-part description are gathered from the user by backward chaining through the rule-based knowledge stored in the system. The user is prompted for each feature and its corresponding attributes. If a value is not selected or entered, then a default value is provided. Each feature and its attributes are verified for correctness by echoing the feature with its attributes to the user.

3.3 SELECTION OF MACHINE AND WORK-HOLDING DEVICE

Several factors affect the selection of a machine that is to be used for machining a component. Since we are considering only turned components in our system, the machines considered are Numerically Controlled (NC) machines, and Single Spindle Automats (SSA). The selection of a

machine depends on the work-holding device and the raw-stock type available. As already discussed above, the work-holding device can be either a chuck or a collet and the raw-stock can be bar stock, cast or semi-finished stock.

Selection of a machine and work-holding device are inter-related and several contradictory conditions occur based on what options are available. Decision making in such a situation is best handled by backward chaining and rule-based knowledge.

For selecting a machine and a work-holding device, rules similar to the ones below are used.

- * IF a single spindle automat is not available,
THEN an NC machine is to be used.
- * IF the stock is BAR and a single spindle automat is
available
THEN the single spindle automat is to be
used.
- * IF the stock is cast or semi-finished and an NC
machine is available
THEN the NC machine is to be used.
- * IF the machine is NC and chuck is available
THEN a chuck is to be used for work-holding.
- * IF the machine is a single spindle automat and a
collet is available and the stock is bar

THEN a collet is to be used.

3.4 SELECTION OF A HOLDING SURFACE

Selection of a work-holding surface depends largely on the tolerance requirements specified by the designer. The tolerance that could be obtained in machining is affected by:

- * The feed rate
- * The depth of cut
- * The slenderness ratio
- * The tool moment of inertia
- * The tool modulus of elasticity
- * The work-piece moment of inertia
- * The work-piece modulus of elasticity
- * The work-piece hardness

The above factors are easily calculated from established formulae and hence formula-based knowledge is used for finding the holding surface.

The following procedure is used to select a holding surface:

1. Initially, the raw-part is assumed to be held at the first possible step from the left side.
2. The cutting force is calculated from the following formula derived by Kronenberg [8].

$$P = C_p * (1000 * A)^{0.803} * (G/5)^{0.16} / 1000 \quad (3.1)$$

where

- P is the cutting force in lb,
- C_p is a cutting force constant depending on the Brinell Hardness Number (BHN) and the tool lip angle (the value of C_p is obtained from a database),
- A is the chip cross-sectional area in inch^2 and is equal to (feed rate)*(depth of cut), and
- G is the slenderness ratio of the work-piece and is equal to (work-piece overhang)/(average diameter),

3. Calculate the tool deflection [3]

$$T_d = P * L_t^3 / (3 * E_t * I_t) \quad (3.2)$$

where

- T_d is the tool deflection in inches
- P is the cutting force in lb.
- L_t is the tool overhang in inches
- E_t is the modulus of elasticity of the tool in psi,
and
- I_t is the moment of inertia of the tool in $(\text{inch})^4$.

4. Calculate the work-piece deflection [3]

$$d_w = P * L_w^3 / (3 * E_w * I_w) \quad (3.3)$$

where

- d_w is the work-piece deflection in inches
- P is the cutting force in lb.

L_w is the work-piece overhang in inches

E_w is the modulus of elasticity of the work-piece in psi, and

I_w is the moment of inertia of the work-piece in (inch)⁴.

5. Calculate the diametrical error [3]

$$d_e = (d_w + d_t)^2 / L_t \quad (3.4)$$

where

d_e is the diametrical error in inches.

6. Calculate the diametrical tolerance [3]

$$d_{tol} = mc_e + d_e \quad (3.5)$$

where

d_{tol} is the diametrical tolerance in inches

mc_e is the machine tool error in inches

d_e is the diametrical error in inches.

As soon as the diametrical tolerance is known, it is compared with the specified design tolerance. If the calculated value is less than the specified value, the assumed initial holding surface is selected. Otherwise, the work-piece is assumed to be held at the next possible step and the procedure is repeated.

3.5 SELECTION OF OPERATION

A rotational part can be machined in an NC turning center or a single spindle automat in different sequences.

The objective of an operations plan is to come up with a sequence that is feasible as well as efficient. The rules used for determining the sequence should be as general as possible to account for the various problems that might be encountered.

The selection of an operation is largely based on the current part description and the finished part description. A large number of qualitative and often conflicting conditions are encountered during the decision-making process. Rule-based knowledge is therefore most appropriate for performing the selection process.

The following general procedure is incorporated in the knowledge base :

1. The length of the raw-part and finished part are compared and facing is done if required.
2. All steps are machined before tapers, threads, fillets, grooves and chamfers. Maximum diameters of tapers and threads are considered to be steps and machined. The step with the largest diameter is machined first, the next largest one next and so forth. Wherever possible, steps are combined into a single cut. After each cut, the current-part description and the finished-part description are compared to check for completion of the machining sequence.
3. All tapers are cut next.

4. Threads are machined next, if this will not affect holding after reversal.
5. Chamfers, fillets and grooves are machined.
6. If no more operations are possible with the current holding, the part is reversed and the procedure is repeated.
7. If more than one reversal is required, then an error message is printed and the consultation is terminated.
8. If the operation is internal, then checks are made to ensure the tool access to the surface to be machined. If a potential access problem is detected, then an error message is printed.

3.6 CALCULATION OF SPEED

As soon as the feature to be machined is decided, the speed of rotation of the work-piece is calculated. The surface speed of rotation of the work-piece is determined by the work-piece material and the tool material. Data-based knowledge simplifies the surface speed determination. The corresponding rotational speed of the work-piece is calculated from the surface speed and feature dimensions using the following relationship:

$$S = 3.82 * \text{SFPM} / D \quad (3.6)$$

where

S is speed of rotation in RPM

SFPM is speed in surface feet per minute

D is the maximum diameter of the feature to be

machined.

3.7 MODIFICATION OF CURRENT PART DESCRIPTION

As soon as each operation is decided, the current part description needs to be modified to reflect the completion of the operation. This is important since it helps us keep track of the previous decisions and also facilitates the use of the modified part description as input for the selection of the next operation.

For the modification of the part description, rule-based knowledge with forward and backward chaining has been implemented. The following conditions are considered during this modification:

1. The operation may span over several design features or be contained within a single feature on the raw-part. Operations spanning several features are termed multi-span and those contained within a single feature are termed single-span. AN example of a multi-span operation is one beginning on a STEP and ending on a TAPER in the current part. An example of a single-span operation is one beginning and ending on the same STEP or TAPER.
2. The operation may begin on any feature and end on any feature on the current part.

The process of modification can be easily explained through an example. Consider the part shown in Figure 3.1.

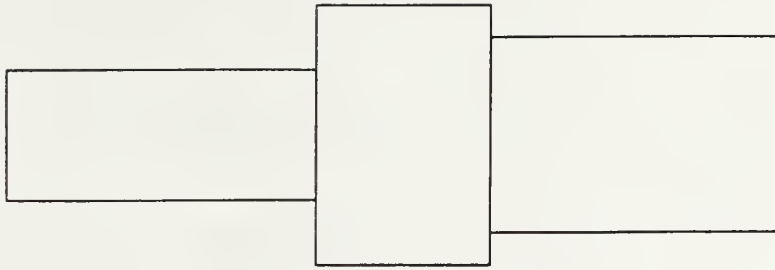


Figure 3.1 Example for Part Modification

The part description is (STEP 0.002 30 0. 2. 1.5 1.5 0. STEP 0.003 25. 2. 3. 2. 2. 0. STEP 0.001 20. 3. 5. 1.75 1.75 0.). Let the operation on this part be a step of diameter 1.0 starting at 1.5 and ending at 3.5. The operation description for this operation is (STEP ,975.25 30. 1.5 3.5 1. 1. 0.). After the operation, the part description is (STEP 0.002 30 0. 1.5 1.5 1.5 0. STEP 0.002 30. 1.5 3.5 1. 1. 0. STEP 0.001 20. 3.5 5. 1.75 1.75 0.).

The modification process by which the new part description is obtained in this example is as follows:

1. Identify the starting and ending features on the part where the operation begins and ends. In the example, the operation starts on the first feature and ends on the third feature.
2. Eliminate the inter-lying features, if any, from the raw-part description. In the example, the second feature is completely overlapped and hence is deleted.
3. Find the intersections of the operation starting and

ending coordinates on the current part. This depends on the type of feature of the current part and the operation being done on the feature. The various possible cases of intersections have been discussed above. In the example, a STEP is cut which begins on a STEP and ends on a STEP on the current part.

4. Split the current part description at the intersections. In the example, the first feature is modified so as to reflect the new operation. The modified feature is (STEP 0.002 30 0. 1.5 1.5 1.5 0.). The last feature becomes (STEP 0.001 20. 3.5 5. 1.75 1.75 0.).

5. Insert the operation description between the starting and ending features. Thus, the modified part description in our example is (STEP 0.002 30 0. 1.5 1.5 1.5 0. STEP 0.002 30. 1.5 3.5 1. 1. 0. STEP 0.001 20. 3.5 5. 1.75 1.75 0.).

Another method through which part modification can be accomplished is given below :

While the part description is not an empty list do :

 If the operation description list is empty then return
 the part description as the modified part;

Else

 if the operation begins in the first feature of
 the part then return the first feature;

 else

 define a new part description by deleting the


```

        first feature of the current part
        description;

    Endif.

    Modify the feature in which the operation begins
        to reflect the current operation.

    If the operation terminates within this feature
        then exit;

    else modify the operation description to be the
        part of the operation lying outside this
        feature and recurse with the modified part
        description and modified operation
        description;

    Endif.

Endif.

Endwhile.

```

The preceding procedure is efficient and compact because of its recursive nature. The semantics and syntax for the above procedure are presented later in this chapter. Due to difficulties encountered with recursion in the software package used, the first procedure has been used in the implementation.

3.8 CALCULATION OF FEED RATE

The feed rate of each operation needs to be calculated and incorporated in the feature description list for each operation. The following formula is used for calculating

feed rates [10]:

$$SF = K * F^2 / RNOSE$$

where

SF is the surface finish in micro inches

F is the feed rate in inches

RNOSE is the tool nose radius in inches and

K is a constant of proportionality and is equal to

$$4.16667 * 10^4$$

The calculated feed rate is compared with the maximum feed rate allowed for the work-piece material and the lower of the two is selected as machining feed rate. This feed rate replaces the surface finish field of the feature description in the operation sequence list.

3.9 OUTPUT

The user can select the output device to be used from three options : SCREEN, PRINTER and FILE. One or more of the three can be chosen and formatted output is directed to the device(s) chosen. Due to the multiple choices available, rule-based knowledge with backward chaining is made use of for printing output to the screen or file or printer or any combination thereof.

3.10 SEMANTIC MODELS

Denotational semantics is a methodology for giving mathematical meaning to programming languages and systems

[11]. Syntax and semantics are characteristics of a programming language. Syntax defines the appearance and structure of the sentences in a programming language. Semantics is the assignment of meaning to the sentences.

The Backus-Naur form (BNF) is widely used for defining syntax. A BNF definition provides valuable documentation that can be used by a programmer with minimal training.

Semantic definition methods are valuable to implementors and programmers, for they provide:

1. A precise standard for a computer implementation.

The standard guarantees that the language is implemented exactly the same on all machines.

2. Useful user documentation.
3. A tool for design and analysis.

Semantics provide a means of formalizing the structure and also as a check for a program. Semantics require syntactically correct programs. Denotational semantics map a program directly to its meaning, called its denotation. The denotation is usually a mathematical value, such as a number or a function.

In our system, the modification of the raw-part involves recursive function definitions and resembles a program structure. The syntax and semantics required for the raw-part modification and for determining the holding

surface-number were formulated for the knowledge base. BNF is used for defining the syntax. A formal syntax and denotational semantics with semantic algebras and valuation functions was developed for the system and is presented below. The standard λ -notation is used for the specification of functions. Overloading is used on some of the operators in the semantic algebras to obtain a more compact semantic model.

3.10.1 Semantics for Part Modification

Abstract syntax:

I \in Modification-instance

P \in Part

O \in Operation

R \in Real number

N \in Natural number

A \in Atom

I ::= modify P by O

P ::= (STEP tol sur-fin start-x end-x diam dum1 dum2) |
 (TAPER tol sur-fin start-x end-x dia1. dia2.
 dum1) | (THREAD tol sur-fin start-x end-x diam.
 pitch form) | (FILLET tol sur-fin start-x radius
 dum1 dum2 dum3) | (GROOVE tol sur-fin start-x
 radius dum1 dum2 dum3) | (CHAMFER tol sur-fin
 start-x angle dum1 dum2 dum3) | P1;P2

O ::= (STEP tol sur-fin start-x end-x diam. dum1 dum2)

```

| (TAPER tol sur-fin start-x end-x diam1. diam2.
dum1) | (THREAD tol sur-fin start-x end-x diam.
pitch form) | (FILLET tol sur-fin start-x radius
dum1 dum2 dum3) | (GROOVE tol sur-fin start-x
radius dum1 dum2 dum3) | (CHAMFER tol sur-fin
start-x angle dum1 dum2 dum3)

```

Valuation functions:

I:Modification-instance -> Part

I[modify P by O] = MODIFY(P,O)

Semantic algebras:

I. Real numbers

Domain $r \in \text{Real}$

Operations:

CP, FEED, DEPTH, TLO, TLINR, TLM, WPINR, WPM, MCERR -> Real

$+, -, *, /$, expt : Real X Real -> Real

log : Real -> Real

All real numbers -> Real

Note: All operators are infixes except for log.

II. Natural numbers

Domain $n \in \text{Nat}$

Operations:

0, 1, 2 -> Nat

$+, -, *, /$: Nat X Nat -> Nat

III. Truth values

Domain $t \in Tr$

Operations:

true, false : Tr

= , > , < , >= , <= , <> $\rightarrow Real \times Real \rightarrow Tr$

IV. Lists

Domain $l \in List$

Operations:

car : List $\rightarrow Atom$

cdr : List $\rightarrow List$

cons : Atom $\times List \rightarrow List$

LIST-REF: List $\times Nat \rightarrow Atom$ [returns n+1th. item of a list]

LIST-REF = $\lambda(l,n). n = 0$

$\rightarrow car(l)$

$\square LIST-REF(cdr(l), n-1)$

FOURTH: List $\rightarrow Real$ [returns 4th. item of a list]

FOURTH = $\lambda(l). LIST-REF(l, 3)$

FIFTH: List $\rightarrow Real$ [returns 5th. item of a list]

FIFTH = $\lambda(l). LIST-REF(l, 4)$

SIXTH: List $\rightarrow Real$ [returns 6th. item of a list]

SIXTH = $\lambda(l). LIST-REF(l, 5)$

SEVENTH: List $\rightarrow Real$ [returns 7th. item of a list]

SEVENTH = $\lambda(l). LIST-REF(l, 6)$

LAST: List \rightarrow Atom [returns last item of a list]

LAST = $\lambda(l). \text{cdr}(l) = \text{nil}$

$\rightarrow \text{car}(l)$

$\square \text{LAST}(\text{cdr}(l))$

LENGTH: List \rightarrow Nat [returns number of atoms in a list]

LENGTH = $\lambda(l). \text{car}(l) = \text{nil}$

$\rightarrow 0$

$\square \rightarrow \text{One plus LENGTH}(\text{cdr}(l))$

LIST-TAIL: List X Nat \rightarrow List [returns last n items of a
list]

LIST-TAIL = $\lambda(l,n). n \geq \text{LENGTH}(l)$

$\rightarrow l$

$\square \text{LIST-TAIL}(\text{cdr}(l), n)$

REVERSE: List \rightarrow List [reverses a list, atom by atom]

REVERSE = $\lambda(l). l = \text{nil}$

$\rightarrow l$

$\square \text{cons}(\text{LAST}(l), (\text{REVERSE}(\text{cdr}(l))))$

GET_TOP_LIST: List X Nat \rightarrow List [returns top n items of a
list]

GET_TOP_LIST = $\lambda(l,n). n = 0$

$\rightarrow \text{nil}$

$\square \text{cons}(\text{car}(l), \text{GET_TOP_LIST}(\text{cdr}(l), n-1))$

APPEND: List X List \rightarrow List [concatenates two lists]

APPEND = $\lambda(l1,l2). \text{car}(l1) = \text{nil}$

$\rightarrow l2$

$\square \text{APPEND}(\text{GET_TOP_LIST}(l1, \text{LENGTH}(l1)-1),$
 $\text{cons}(\text{LAST}(l1), l2))$

CHANGE: List X Nat X Atom \rightarrow List [changes nth item of list
with atom]

CHANGE = $\lambda(l,n,a). \text{APPEND}(\text{GET_TOP_LIST}(l,n), (\text{cons}(a,$
 $\text{LIST-TAIL}(l,n))))$

V. Operation

Domain o \in Operation = List

Operations:

car : Operation \rightarrow Atom

cdr : Operation \rightarrow List

LIST-REF: Operation X Nat \rightarrow Atom [returns nth. item of an
operation]

FOURTH: Operation \rightarrow Real [returns 4th. item of an opn.]

FIFTH: Operation \rightarrow Real [returns 5th. item of an opn.]

SIXTH: Operation \rightarrow Real [returns 6th. item of an opn.]

SEVENTH: Operation \rightarrow Real [returns 7th. item of an opn.]

CHANGE: Operation X Nat X Atom \rightarrow Operation [changes the nth
item of opn. with atom]

VI. Part

Domain $p \in \text{Part} = \text{List}$

Operations:

$\text{car} : \text{Part} \rightarrow \text{Atom}$

$\text{cdr} : \text{Part} \rightarrow \text{List}$

$\text{LIST-REF} : \text{Part} \times \text{Nat} \rightarrow \text{Atom}$ [returns nth. item of a part]

$\text{FOURTH} : \text{Part} \rightarrow \text{Real}$ [returns 4th. item of a part]

$\text{FIFTH} : \text{Part} \rightarrow \text{Real}$ [returns 5th. item of a part]

$\text{SIXTH} : \text{Part} \rightarrow \text{Real}$ [returns 6th. item of a part]

$\text{SEVENTH} : \text{Part} \rightarrow \text{Real}$ [returns 7th. item of a part]

$\text{LIST-TAIL} : \text{Part} \times \text{Nat} \rightarrow \text{List}$ [returns last n items of a
list]

$\text{REVERSE} : \text{Part} \rightarrow \text{Part}$ [reverses a list, atom by atom]

$\text{GET_TOP_LIST} : \text{Part} \times \text{Nat} \rightarrow \text{List}$ [returns top n items of a
list]

$\text{APPEND} : \text{List} \times \text{List} \rightarrow \text{List} + \text{Part}$ [merges two lists]

$\text{CHANGE} : \text{Part} \times \text{Nat} \times \text{Atom} \rightarrow \text{Part}$ [changes nth item of list
with atom]

$\text{LEAD_PART} : \text{Part} \rightarrow \text{Part}$ [returns first feature of part list]

$\text{LEAD_PART} = \lambda(p). \text{GET_TOP_LIST}(p, 8)$

TAIL_PART: Part -> Part [returns part list without first
feature]

TAIL_PART = $\lambda(p). \text{LIST-TAIL}(p, 7)$

BEGINS: Part X Operation -> Tr [checks if operation begins
on the first feature of a part]

BEGINS = $\lambda(p, o). \text{FOURTH}(o) \geq \text{FOURTH}(p)$ and
 $\text{FOURTH}(o) < \text{FIFTH}(p)$
-> true
 \square false

SPLIT1: Part X Operation -> Part [returns the part modified
by the operation lying within
the first feature]

SPLIT1 = $\lambda(p, o). \text{FIFTH}(p) < \text{FIFTH}(o)$
->inPart(CHANGE(CHANGE(CHANGE(CHANGE
(CHANGE(CHANGE(p, 1, LIST-REF(o, 0)),
2, LIST-REF(o, 1)),
3, LIST-REF(o, 2)),
4, FOURTH(o)),
6, SIXTH(o)),
7, SEVENTH(o)))
 \square ->inPart(CHANGE(CHANGE(CHANGE(CHANGE(CHANGE
(CHANGE(CHANGE(p, 1, LIST-REF(o, 0)),
2, LIST-REF(o, 1)),
3, LIST-REF(o, 2)),

```

4,FOURTH(o)),
5,FIFTH(o))
6,SIXTH(o)),
7,SEVENTH(o)))

```

SPLIT2: Part X Operation -> Operation [returns the operation
lying outside the first feature]

```

SPLIT2 = λ(p,o).FIFTH(p) < FIFTH(o)
        -> inOperation(CHANGE(o,4,FIFTH(p)))
        □ nil

```

MODSTEP: Part X Operation -> Part [modifies part if
operation is a step]

```

MODSTEP = λ(p,o).inPart(CHANGE(p,5,FOURTH(o)))

```

MODTAPER: Part X Operation -> Part [modifies part if
operation is a taper]

```

MODTAPER= λ(p,o).SEVENTH(p) > SIXTH(p)
        -> let D2=(FOURTH(o) - FOURTH(p)) *
            (SEVENTH(p) - SIXTH(p)) / (FIFTH(p)
            - FOURTH(p)) + SIXTH(p)
            in inPart((CHANGE(p,7,D2)))
        □-> let D1=(FIFTH(p) - FIFTH(o)) * (SIXTH(p) -
            SEVENTH(p)) / (FIFTH(p) -
            FOURTH(p)) + SEVENTH(p)
            in inPart((CHANGE(p,6,D1)))

```

MODIFY: Part X Operation -> Part [modifies the part for the
given operation]

MODIFY = $\lambda(p,o).$ p = nil

-> nil

□

{o = nil

-> p

□ let f = LEAD_PART(p)

in

(BEGINS(f,o)->

CASE car(f)

of STEP ->

APPEND(MODSTEP(f,o),

APPEND(SPLIT1(f,o),

MODIFY(TAIL_PART(p),

SPLIT2(f,o)))

of TAPER ->

APPEND(MODTAPER(f,o),

APPEND(SPLIT1(f,o),

MODIFY(TAIL_PART(p),

SPLIT2(f,o)))

of (CHAMFER|FILLET|GROOVE) ->

f

ENDCASE

□ APPEND(LEAD_FEATURE(p),

MODIFY(TAIL_PART(p),o) }

3.10.2 Semantics for Holding Surface Selection

Abstract syntax:

S \in Surface-number evaluation

P \in Part-list

R \in Real number

N \in Natural number

A \in Atom

S ::= surface_number P

P ::= (STEP R1 R2 R3 R4 R5 R6 R7) | (TAPER R1 R2 R3 R4
R5 R6 R7) | (THREAD R1 R2 R3 R4 R5 R6 A1) |
(FILLET R1 R2 R3 R4 R5 R6 R7) | (GROOVE R1 R2 R3
R4 R5 R6 R7) | (CHAMFER R1 R2 R3 R4 R5 R6 R7) |
P1;P2

Valuation functions:

S:[surface_number P] = FSN(P,0)

Semantic algebra:

I. Natural numbers - Same as before

II. Real numbers - Same as before with the following
additional operations:

CUT_FORCE: Real \rightarrow Real [returns cutting force of tool]

CUT_FORCE = $\lambda(s1).((CP * 1000) * ((FEED * DEPTH) \text{ expt } 0.803)$
 $* ((s1 / 5.0) \text{ expt } 0.16)) / 1000$

TOOL_DEFL: Real -> Real [returns tool deflection]

TOOL_DEFL = $\lambda(cf).(cf * (TLO \text{ expt } 3.0)) / (3. * TLINR * TLM)$

WORK_DEFL: Real X Real -> Real [returns work-piece
deflection]

WORK_DEFL = $\lambda(cf,wo).(cf * (wo \text{ expt } 3.)) / (3. * WPINR * WPM)$

DIAERROR: Real X Real -> Real [returns diametrical error]

DIAERROR = $\lambda(tdf,wdf).((wdf + tdf) \text{ expt } 2) / TLO$

III. Truth values - Same as before

IV. Lists - Same as before

V. Part - Same as before with the following additional
operations:

Domain $p \in \text{Part} = \text{List}$

Operations:

COMPARE: Part X Real -> Tr [compares the calculated
tolerance with specified tolerance]

COMPARE = $\lambda(p,dtol).(LIST-REF(p,(length(p) - 8)) < dtol$
-> false

□ true

AVDIA2 = $\lambda(p).(car(p) = STEP) \text{ or } (car(p) = TAPER) \text{ or}$
 $(car(p) = THREAD)$
-> $(SIXTH(p) + SEVENTH(p)) / 2.0$

□ zero

AVDIA1: Part -> Real [returns average diameter of a part]

```
AVDIA1 = λ(p).let sum=0
      in (sum = sum + AVDIA2(LEAD_PART(p))
          + AVDIA1(TAIL_PART(p)))
```

AVDIA: Part X Nat -> Real [returns the average diameter of
a part from holding surface]

```
AVDIA = λ(p,n).let p1=RETURN_LIST(p,n)
      in (AVDIA1(p1) / (length(p) / eight))
```

```
MAXX1 = λ(p).(car(p) = STEP) or (car(p) = TAPER) or
          (car(p) = THREAD)
          -> FIFTH(p)
      □ FOURTH(p)
```

MAXX: Part -> Real [returns maximum x coordinate of part]

```
MAXX = λ(p).(max (MAXX1(LEAD_PART(p)),MAXX(p)))
```

```
MINX1 = λ(p).(not nil p) -> FOURTH(p)
      □ 100000
```

MINX: Part -> Real [returns minimum x coordinate of part]

```
MINX = λ(p).(min (MINX1(LEAD_PART(p)),MINX(TAIL_PART(p)))
```

RETURN_LIST: Part X Nat -> List [returns list from holding
surface]

```
RETURN_LIST = λ(p,n).n equals zero
              -> p
              □ RETURN_LIST(TAIL_PART(p),n - 1)
```

```

WPOVR: Part X Nat -> Real [returns the work-piece overhang]
WPOVR =  $\lambda(p,n).$ let p1=RETURN_LIST(p,n)
                                in (MAXX(p1) - MINX(p1))

FSN: Part X Nat -> Nat [returns holding surface number]
FSN =  $\lambda(p,n).$ let ad=AVDIA(p,n)
                wo=WPOVR(p,n)
                in (let sl=(wo / ad)
                    in (let cf=CUT_FORCE(sl)
                        in (let tdf=TOOL_DEFL(cf)
                            wdf=WORK_DEFL(cf,wo)
                            in (let derr=DIAERROR(tdf,wdf)
                                in (let dtol=derr + MCERR
                                    in (not (COMPARE(p,dtol))
                                        -> FSN(TAIL_PART(p),n+1)
                                    □ n))))))

```

The syntax and semantics shown above are for rotational components using a feature-based representation. For integration of CAD and CAM, this definition could be expanded to a more general form to account for other manufacturing operations such as milling, drilling, shaping etc. The denotational semantics definition is very effective and should be developed for future investigations into other manufacturing operations.

CHAPTER 4

KNOWLEDGE-BASE IMPLEMENTATION

The implementation of the knowledge base structure discussed in chapter 3 requires an expert system shell capable of providing the following features:

1. The expert system shell should have a high degree of flexibility.
2. It should provide a natural dialogue interaction.
3. The system should have the capability to handle non-numeric data.
4. It should also provide a frame based structure within the domain.
5. It should support Lisp and database access.

All of the above features and many more are provided by Personal Consultant Plus (PC Plus), which is an expert system shell developed by Texas Instruments. In this chapter, we will give an overview of PC Plus and our knowledge base implementation using PC Plus.

4.1. PC PLUS - AN OVERVIEW

PC Plus is a highly functional tool for developing expert system applications. PC Plus offers a sophisticated frame structure for handling multidimensional problems, techniques for imposing rule-order strategies, the ability to extend and customize the consultation environment, and a large rule capacity.

4.1.1 Features of PC Plus

PC Plus includes many powerful features that made it attractive for our application:

A highly interactive environment for development and testing facilitates the development of the knowledge base in parts. The knowledge base is easily verified using the debugging and value-checking aids provided by the shell. The ability of the system to explain things in English (e.g. why some particular information is needed or how a certain conclusion is reached, etc.) is helpful in determining the path followed during a consultation.

PC Plus provides access to external DOS files and databases created by dBASE II, dBASE III, dBASE III Plus, and Lotus 1-2-3 spreadsheets. This feature is essential to our application since the material data such as SAE number, BHN, speeds and feeds, etc. are easily created and maintained using dBASE III Plus, independent of the knowledge base.

The ability to extend a knowledge base through the use of user-defined Lisp functions is well provided for in PC Plus. Since our knowledge base consists of several lists representing the part and operation descriptions, the use of specialized Lisp functions to manipulate these lists simplifies data handling.

The capability of PC Plus to allow the knowledge base to be divided into logically different, but related segments in the form of frames is advantageous since our knowledge base is made up of segments as discussed in chapter 3. Each of the steps discussed, such as, input information, holding surface determination, feed rate calculation, etc. are implemented as separate frames.

PC Plus also provides additional means (other than inferencing and prompting) such as a parameter's method property or active-value property to evaluate and set parameter values. The active-value property provides a mechanism for a looping structure which is not provided naturally by PC Plus. This looping structure is discussed later in this chapter.

In PC Plus, a knowledge base is structured into frames, parameters, rules, meta-rules, variables, functions and textags.

4.1.2 Frames

A frame is a structure that provides for the collection of rules, parameters, variables and functions required for solving for a goal or a set of goals. The goals are parameters which are assigned values through rules or functions. The frame contains information on how to achieve the goals.

A frame resembles a subroutine in a conventional programming knowledge. That is, each frame performs a specific function.

A knowledge base always contains at least one frame-the root frame. Additional frames or subframes may be created if the knowledge base is large or complex. Each frame has a set of properties which determine its characteristics.

PC Plus instantiates the root frame when a consultation is started by a client. PC Plus does not instantiate subframes automatically. The mechanism to cause and control instantiation must be built into the knowledge base by the developer. When PC Plus instantiates a subframe, it must temporarily put aside the search for the ancestor frame's goals and trace the subframe goal or goals.

PC Plus instantiates a frame during a consultation by one of two methods:

- * Include the CONSIDERFRAME function in the THEN statement of a rule that will fire in the parent frame. The CONSIDERFRAME function takes a frame name as an argument and could also be used in the ACTIVE-VALUE property of a parameter.
- * Structure the knowledge base so that PC Plus must instantiate the subframe to find a value needed in an ancestor frame. With this method, the frame is

instantiated in the normal course of backward chaining.

4.1.3 Parameters

A parameter is a structure that identifies or contains a piece of information that PC Plus uses to arrive at a conclusion. Like frames, parameters have properties that PC Plus may use during a consultation to reach a correct conclusion.

Three properties are prompted for when a parameter is added: TRANSLATION, TYPE and PROMPT. There are other optional properties that can be used to control the manner in which the parameter value is determined and also control other parameters and rule evaluation.

- * TYPE - Determines whether a parameter is single-valued, multivalued, etc. and positive, negative, integer, etc.
- * ACTIVE-VALUE - Specifies an action to be taken whenever the parameter value is set. It could be used set another parameter's value whose ACTIVE-VALUE sets the first parameter's value thereby giving a looping structure.
- * ASKFIRST - Tells PC Plus to ask the client for the parameter's value before using other methods.
- * DEFAULT - Provides a default value for a parameter.
- * METHOD - The parameter is assigned a value or values

returned by a function call.

- * PROMPT - Specifies the phrase that PC Plus uses to ask the client for a parameter value.
- * TRANSLATION - Briefly describes the parameter.

PC Plus follows a specific sequence in tracing a parameter:

1. If the parameter has a PROMPT property and an ASKFIRST property and does not have a previously assigned value with certainty greater than 20, prompt the client.
2. If the parameter has a METHOD property and has no value with certainty 100, use the function specified in the METHOD property.
3. If the parameter appears in the THEN part of any of the rules, try those rules.
4. If the parameter has a PROMPT property but no ASKFIRST property and does not have a previously assigned value with certainty greater than 20, prompt the client.
5. If the parameter has a DEFAULT property and its value has not been determined with certainty greater than 20, use the DEFAULT value.

When this process is complete, the parameter is considered to have been traced.

4.1.4 Variables

A variable is another structure for storing information

in a knowledge base. Variables provide values to which the entire knowledge base has access during a consultation. Variables differ from parameters in four essential ways:

- * PC Plus does not trace variables.
- * Variables cannot have certainty factors.
- * Variables are global to the knowledge base and are not restricted by the rules of frame inheritance.
- * Variables are easily modified as compared to a parameter's value which is difficult to modify once the parameter has been marked as traced.

4.1.5 Rules

PC Plus rules are if-then statements that express the relationships among parameters. PC Plus rules can be written in Lisp or in Abbreviated Rule Language (ARL), a language similar to English. Rules structure knowledge in an IF-THEN form: IF certain conditions are met, THEN a piece of knowledge is true or has a particular value. The IF part can consist of logical comparisons of values of parameters or variables, Lisp functions which return true or false, and YES/NO parameters. The THEN part consists of actions such as assigning values to parameters and variables, and executing specific Lisp functions.

4.1.6 Functions

In PC Plus new Lisp functions can be created or exist-

ing Lisp functions can be used to accomplish a particular purpose in the knowledge base. These functions greatly extend the versatility and flexibility of PC Plus. They extend the capabilities of the knowledge base by performing specialized operations that no PC Plus functions perform, and can replace a calculation or sequence of functions that appears in several places in the knowledge base.

4.2 STRUCTURE OF OUR KNOWLEDGE BASE

The frame based structure provided by Personal Consultant Plus has been made use of in developing our knowledge base. The knowledge base consists of a root frame called PROCESSOR. PROCESSOR has four subframes: INPUT, HOLDING, SURFACE-FINISH, and PRINT-OUTPUT. INPUT is the parent frame for INPUT-LIST. HOLDING has CHECK-STOCK as a subframe which has BAR-STOCK and CAST-STOCK as child frames. BAR-STOCK has its own subframe called MODBAR and CAST-STOCK has MODIFY as its subframe. The outline is shown in the flow chart in Figure 4.1. The path followed through the knowledge base during a consultation is explained below:

At the start of a consultation, PC Plus instantiates the root frame PROCESSOR. PROCESSOR forces instantiation of the INPUT frame by causing a trace for the INPUT frame's goal parameter. Since the structure of the lists used for the part descriptions have an uniform length of eight, to get values for the raw-part, finished part (external) and

finished part (internal) descriptions, the INPUT frame instantiates the INPUT-LIST frame several times. The number of times INPUT-LIST is to be instantiated is controlled by the client's input to prompts. This looping structure is

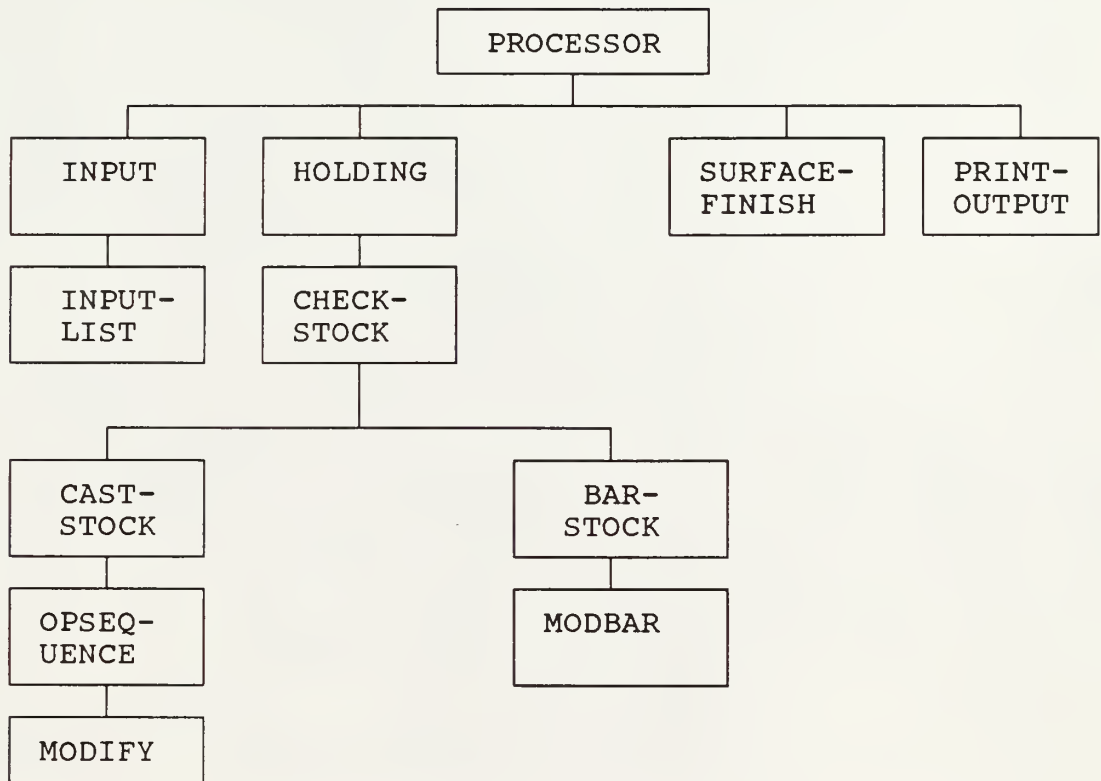


Figure 4.1 Structure of our Knowledge Base.

done by the CONSIDERFRAME function in the THEN part of rules in the INPUT frame. After obtaining the list values, the consultation returns to the PROCESSOR where additional input such as work-piece material, tool material, tool width, etc. are prompted for. Next, PROCESSOR causes instantiation of the HOLDING frame which determines the machine and work-

holding device to be used for machining. HOLDING, in turn instantiates the CHECK-STOCK frame whose function is to instantiate the BAR-STOCK or the CAST-STOCK frame depending on the stock type. If the stock is bar, then BAR-STOCK frame is instantiated. If the stock is cast or semi-finished, then CAST-STOCK is instantiated. The BAR-STOCK frame determines the operation sequence as discussed in chapter 3. After each operation has been selected, BAR-STOCK instantiates the MODBAR frame from ACTIVE-VALUE properties of parameters using the CONSIDERFRAME function. MODBAR modifies the current-part description to reflect the effect of the selected operation completion. The CAST-STOCK frame instantiates the HOLDING-SURFACE frame which determines the holding surface on the part. As soon as the holding surface is determined, the OPSEQUENCE frame is instantiated to determine the operation sequence. As in the case of bar stock, after each operation, the MODIFY frame is instantiated to modify the current part to reflect the effect of the operation. The OPSEQUENCE frame is exited when no more operations are possible with that holding surface. At this stage, control returns to the CHECK-STOCK frame which checks for completion of operation sequence. If some more operations are required to be done to achieve the finished part, then the current part is reversed and the HOLDING-SURFACE is instantiated and the process repeats. If all the required operations are not completed within one reversal, the system

prints an error message. After this, control is transferred back to the PROCESSOR, which instantiates the SURFACE-FINISH frame for the calculation of the feed rate for each operation. After the feed rates are calculated the results are printed from the PRINT-OUTPUT frame instantiated from PROCESSOR. In the following sections, we will be examining the function of each of the frames in detail.

4.2.1 Frame PROCESSOR

PROCESSOR is the root frame of the knowledge base system. Its function is to get all the initial input required for the system and to control the flow of consultation through the system. The processor gets data for the final part description (external and internal) and the raw part description, along with other input parameters discussed in chapter 3.

4.2.2 Frame INPUT

This frame is instantiated from the PROCESSOR frame to get input for the raw part, finished part (external), and finished part (internal) descriptions. This frame is instantiated by normal backward chaining for the goal from PROCESSOR. The user is prompted to enter the finished part description (external only) for the subsequent prompts. The INPUT-LIST frame is instantiated to get the features and their attributes. INPUT-LIST is instantiated through back-

ward chaining for its goal. This process is repeated until the user chooses to end the input for the finished part description (external only). The same process is used for finished part (internal) and raw part descriptions.

4.2.3 Frame INPUT-LIST

The purpose of this frame is to present the user with a list of available features, such as, STEP, TAPER, THREAD, FILLET, GROOVE, CHAMFER, HOLE, ITAPER (internal taper), and ITHREAD (internal thread). As soon as the user selects a feature, this frame prompts for the associated attributes of the feature. Upon completion of the input, the list is presented to the user for verification.

4.2.4 Frame HOLDING

This frame determines the machine to be used and the work-holding device for that machine for the given raw stock. It is instantiated from PROCESSOR through backward chaining. HOLDING performs backward chaining and forward chaining to determine its goals. Some of the rules used in this frame (in ARL) are:

RULE042

IF || ((STOCK = CAST OR STOCK = SEMI-FINISHED) AND
 NCM)

THEN || MC = NC

TRANSLATION || If stock material is either cast or semi-finished and a NC machine is available, then the machine to

be used is an NC machine.

RULE044

IF || (MC = SS-AUTOMAT AND COLLET)

THEN || (WHD = COLLET AND DGOAL)

TRANSLATION || If machine to be used is a single spindle automat and collet is available, then the work-holding device is a collet and the dummy goal is set to true.

4.2.5 Frame CHECK-STOCK

As soon as the machine and work-holding device are known, the consultation needs to make a decision on the frame to instantiate for different types of stock material. For this, the frame CHECK-STOCK is provided. This frame checks for the raw-stock material type and if it is BAR, then it causes instantiation of the frame BAR-STOCK. If the stock is CAST or SEMI-FINISHED, then CAST-STOCK is instantiated.

4.2.6 Frame BAR-STOCK

This frame is instantiated if, and only if, the raw material is a bar stock. For bar stocks, the holding surface need not be determined. The sequence of operations is determined in this frame with the methodology discussed in chapter 3. Control throughout this frame is established by the ACTIVE-VALUE properties of parameters. For example, since steps are to be machined before any other operation, a

parameter called STEPMOD is created. This parameter has an ACTIVE-VALUE property which updates the operation sequence list and also modifies STEPMOD if there are any steps to be machined. This creates the required looping structure for determining the next operation. The same concept is used for other features also. For internal operations, checks are made to check for tool access to the surface to be machined. If a possible access problem arises, then a warning message is printed.

The BAR-STOCK frame instantiates the MODBAR frame after each operation is decided. MODBAR is instantiated several times, once for each operation.

4.2.7 Frame MODBAR

MODBAR is used to modify the raw-part description for the given operation so as to reflect the latest changes to the part. The part description is modified taking into account the various conditions that occur as discussed in chapter 3. The semantic model developed in chapter 3 is used as the basis for this frame.

4.2.8 Frame CAST-STOCK

This frame is instantiated only if the stock material is CAST or SEMI-FINISHED. The function of this frame is to provide a controlling structure for looping through the HOLD-SURFACE and OPSEQUENCE frames. This frame causes

instantiation of the HOLD-SURFACE frame which, in turn instantiates the OPSEQUENCE frame. These two frames are discussed in the next two sections.

4.2.9 Frame HOLD-SURFACE

HOLD-SURFACE determines the holding surface on the part, giving due consideration to the design tolerances. This is accomplished by :

- * Selecting a temporary holding surface and calculating the tolerances for that holding.
- * If the tolerances are not satisfied, then a new surface is selected and tolerances are calculated again.
- * If the tolerances are satisfied, then that surface becomes the holding surface.

The formulae used are given in chapter 3. The holding surface number is inherited by the OPSEQUENCE sub-frame which is then instantiated.

4.2.10 Frame OPSEQUENCE

OPSEQUENCE determines the sequence of operations for CAST or SEMI-FINISHED raw stock material. The methodology for this has been explained clearly in chapter 3. The length of the raw part and finished part are compared and the facing operation is appended to the operation sequence list if necessary. In order to generate the operation sequence several parameters (STEPMOD, TAPERMOD, THREADMOD,

etc.) with ACTIVE-VALUE property are created. Initially, since steps are to be machined first, the STEPMOD parameter is given a suitable value. At this point, its ACTIVE-VALUE property, which contains the logic for determining the step to be machined, is executed. Similar methodology is used for determining the complete operation sequence. For internal operations, checks are made to check for tool access to the surface to be machined. If a possible access problem arises, then a warning message is printed. As soon as all operations possible with that holding are done, control is transferred back to the CAST-STOCK frame which reverses the part (i.e., its description) and loops again through the HOLD-SURFACE and OPSEQUENCE frames.

As was the case with the BAR-STOCK frame, the raw-part is modified after each operation is known. This is accomplished in the MODIFY frame.

4.2.11 Frame MODIFY

The MODIFY frame is used to modify the part description for the given operation. This frame makes use of the same rules used by the MODBAR frame and hence shares its rule-group with the MODBAR frame.

4.2.12 Frame SURFACE-FINISH

Once the sequence of operations is known, the root frame instantiates the SURFACE-FINISH frame. This frame

calculates the feed rate for each operation to satisfy surface finish requirements. The calculations are done using a Lisp function. The calculated value is included in the operation sequence list. The feed rate is calculated as discussed in chapter 3.

4.2.13 Frame PRINT-OUTPUT

This frame is instantiated at the end of the consultation from the root frame. The purpose is to print the results in an easy to understand format to the screen, printer or file. The user is given the option of selecting the output devices into which the results are to be written. The user can select one or more of the available output devices.

CHAPTER 5

RESULTS

The developed knowledge-based system for process planning of turned components was tested to ensure the correctness of the implementation. To check the system, several examples of varying complexity were used. The examples range from stepped shafts to components with multiple tapers and threads. Examples included components with external features only and components with both external and internal features.

The first example was made up of steps only. In the second example, a taper was added. The third example was constructed with two tapers and several steps. The fourth example was made a little bit more complex with steps, a thread, a fillet, a groove and a chamfer. Example five was formulated to verify that the system selects the correct sequence of operations for internal features. Holes and an internal taper were included in the fifth example. In the sixth example, an unreachable hole was specified in the input to check whether the system recognized the existence of a tool access problem. A cone clutch, a typical application problem, was selected for the seventh example. The eighth example was to test the system with multiple features such as tapers and threads. Several other examples were run on the system and reasonable process plans were generated in

all cases. The output from the examples along with the figures are given in the rest of this chapter. For all examples the figures after each operation have also been given for better visualization. For better understanding of the features and their attributes in the following examples, the tables for interpreting design and manufacturing features presented in chapter 2 are repeated below.

Feature Class	Attributes							
	1	2	3	4	5	6	7	8
Step	STEP	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dummy	Dummy
Taper	TAPER	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Thread	THREAD	Tol.	Sur.Fin.	X-Start	X-End	Dia.	Pitch	Form
Fillet	FILLET	Tol.	Sur.Fin.	X-Start	Radius	Dummy	Dummy	Dummy
Groove	GROOVE	Tol.	Sur.Fin.	X-Start	Radius	Dummy	Dummy	Dummy
Chamfer	CHAMFER	Tol.	Sur.Fin.	X-Start	Angle	Dummy	Dummy	Dummy
Hole	HOLE	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Taper	ITAPER	Tol.	Sur.Fin.	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Thrd.	ITHREAD	Tol.	Sur.Fin.	X-Start	X-End	Dia.	Pitch	Form

Table 5.1 Design Feature Descriptors and Attributes.

Feature Class	Attributes							
	1	2	3	4	5	6	7	8
Step	STEP	Speed	Feed	X-Start	X-End	Dia.1	Dummy	Dummy
Taper	TAPER	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Thread	THREAD	Speed	Feed	X-Start	X-End	Dia.	Pitch	Form
Fillet	FILLET	Speed	Feed	X-Start	Radius	Dummy	Dummy	Dummy
Groove	GROOVE	Speed	Feed	X-Start	Radius	Dummy	Dummy	Dummy
Chamfer	CHAMFER	Speed	Feed	X-Start	Angle	Dummy	Dummy	Dummy
Hole	HOLE	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Taper	ITAPER	Speed	Feed	X-Start	X-End	Dia.1	Dia.2	Dummy
Int.Thrd.	ITHREAD	Speed	Feed	X-Start	X-End	Dia.	Pitch	Form

Table 5.2 Manufacturing Feature Descriptors and Attributes.

Example 1

INPUT

Raw part description :

(STEP 0.001 100 0. 3. 2.6 2.6 0

STEP 0.001 100 3. 7. 2.8 2.8 0)

Finished part description :

(STEP 0.001 20 0. 2. 1.5 1.5 0

STEP 0.001 30 2. 4. 2.5 2.5 0

STEP 0.001 25 4. 4.75 2. 2. 0

STEP 0.001 20 4.75 6.75 2.5 2.5 0)

Internal part description : ()

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.2

Width of tool : 1.1

Tool material : COBALTHSS

Tool over-hang : 2.

Tool lip-angle : 80

Stock material : CAST

Machine tool error : 1.e-4

Work-piece SAE number : C1020

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.

FACE 0.25

STEP 204.642857142857 0.025 2. 6.75 2.5 2.5 0

STEP 229.2 0.0229128803078097 4. 4.75 2. 2. 0

REVERSE OBJECT

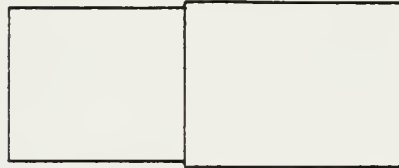
HOLD THE OBJECT BEGINNING AT 4.75 AND ENDING AT 6.75

STEP 220.384615384615 0.0204939031714315 0. 2. 1.5 1.5 0)

RAW PART



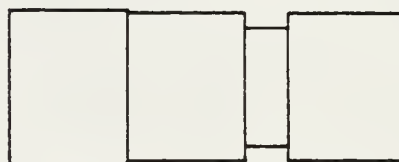
HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.
FACE 0.25



STEP 204.642857142857 0.025 2. 6.75 2.5 2.5 0



STEP 229.2 0.0229128803078097 4. 4.75 2. 2. 0



REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 4.75 AND ENDING AT 6.75

STEP 220.384615384615 0.0204939031714315 0. 2. 1.5 1.5 0

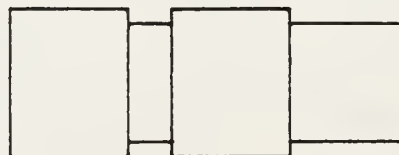


Figure 5.1 Operation Sequence for Example 1.

Example 2

INPUT

Raw part description :

 (STEP 0.001 100 0. 4. 2.5 2.5 0

 STEP 0.01 100 4. 8. 2.2 2.2 0)

Finished part description :

 (STEP 0.001 20 0. 2. 1.5 1.5 0

 STEP 0.001 30 2. 4. 2. 2. 0

 TAPER 0.001 25 4. 6. 2. 1. 0

 STEP 0.001 20 6. 7.5 1. 1. 0)

Internal part description : ()

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.2

Width of tool : 1.4

Tool material : CAST-ALLOY

Tool over-hang : 2.

Tool lip-angle : 65

Stock material : SEMI-FINISHED

Machine tool error : 1.e-4

Work-piece SAE number : C1022

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.

FACE 0.5

STEP 397.28 0.0189736674789037 2. 7.5 2. 2. 0

STEP 496.6 0.0154919346241845 6. 7.5 1. 1. 0

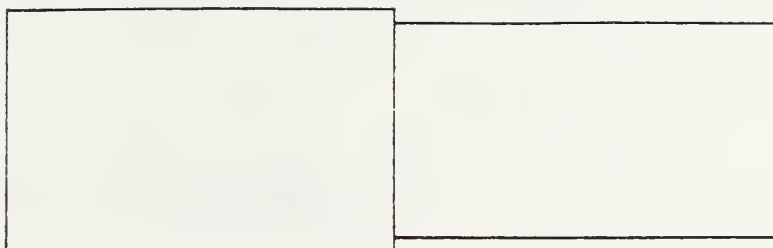
TAPER 496.6 0.0173205094613296 4. 6. 2. 1. 0

REVERSE OBJECT

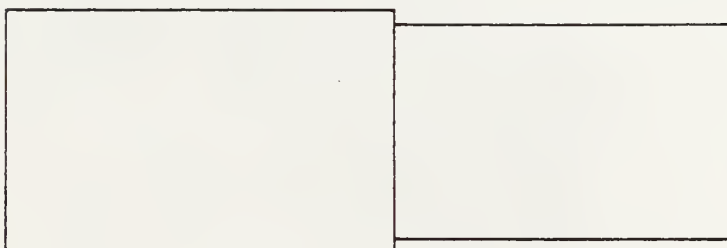
HOLD THE OBJECT BEGINNING AT 6. AND ENDING AT 7.5

STEP 397.28 0.0154919346241845 0. 2. 1.5 1.5 0)

RAW PART



HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.
FACE 0.5



STEP 397.28 0.0189736674789037 2. 7.5 2. 2. 0

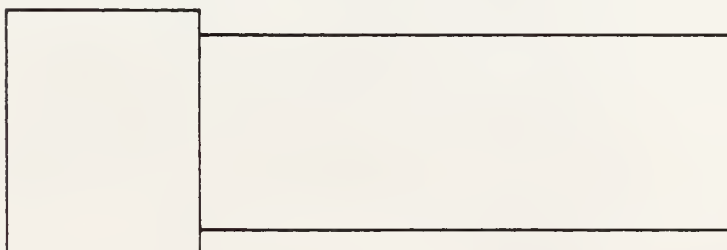
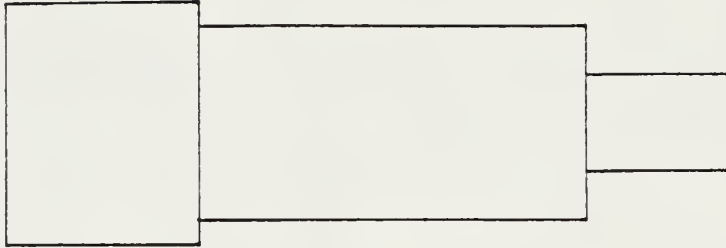
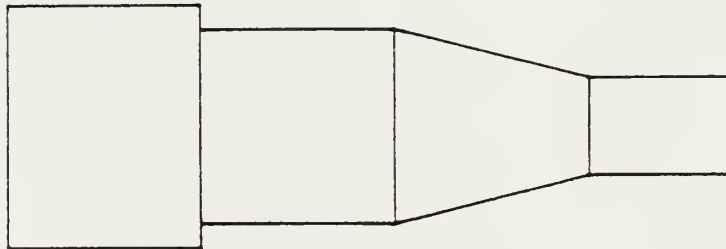


Figure 5.2a Operation Sequence for Example 2.

STEP 496.6 0.0154919346241345 6. 7.5 1. 1. 0



TAPER 496.6 0.0173205094613296 4. 6. 2. 1. 0



REVERSE OBJECT
HOLD THE OBJECT BEGINNING AT 6. AND ENDING AT 7.5
STEP 397.28 0.0154919346241845 0. 2. 1.5 1.5 0

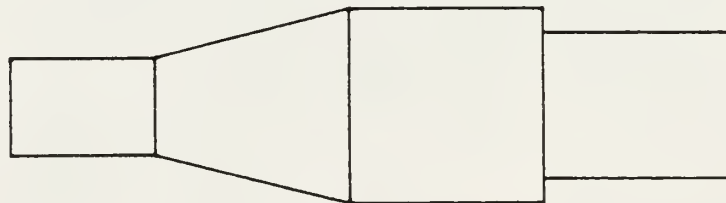


Figure 5.2b Operation Sequence for Example 2.

Example 3

INPUT

Raw part description :
(STEP 0.001 100 0. 9.3 3.4 3.4 0)
Finished part description :
(STEP 0.001 40 0. 2.5 1.5 1.5 0
TAPER 0.001 50 2.5 4. 1.5 3. 0
STEP 0.001 45 4. 5. 1.5 1.5 0
TAPER 0.02 50 5. 6.5 3. 1.5 0
STEP 0.001 40 6.5 9. 1.5 1.5 0)
Internal part description : ()
Machines available : ((NC 100) (SS-AUTOMAT 100))
Holding devices available : ((CHUCK 100) (COLLET 100))
Breadth of tool : 1.3
Width of tool : 1.4
Tool material : SINTERED-CARBIDE
Tool over-hang : 2.
Tool lip-angle : 75
Stock material : SEMI-FINISHED
Machine tool error : 1.e-4
Work-piece SAE number : B1010

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.5

FACE 0.3000000000000001

STEP 978.594117647059 0.028 2.5 9. 3. 3. 0

STEP 1109.073333333333 0.028 4. 5. 1.5 1.5 0

STEP 1109.073333333333 0.026832817876623 6.5 9. 1.5 1.5 0

TAPER 1109.073333333333 0.028 2.5 4. 1.5 3. 0

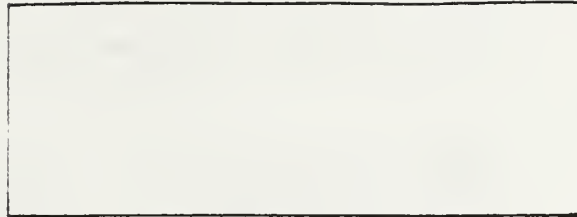
TAPER 1109.073333333333 0.028 5. 6.5 3. 1.5 0

REVERSE OBJECT

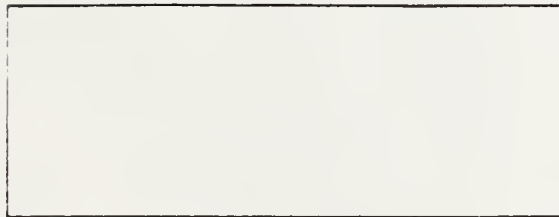
HOLD THE OBJECT BEGINNING AT 6.5 AND ENDING AT 9.

STEP 978.594117647059 0.026832817876623 0. 2.5 1.5 1.5 0)

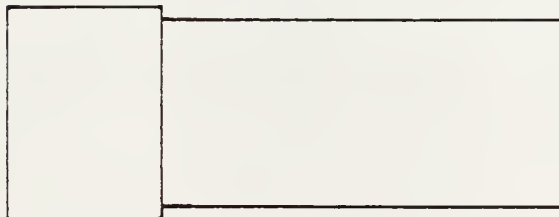
RAW PART



HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.5
FACE 0.3000000000000001



STEP 978.594117647059 0.028 2.5 9. 3. 3. 0



STEP 1109.073333333333 0.028 4. 5. 1.5 1.5 0

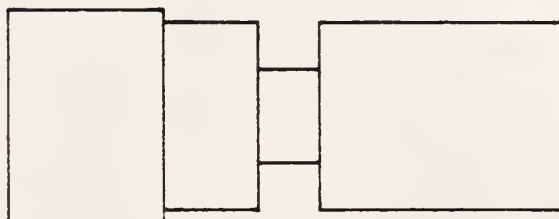
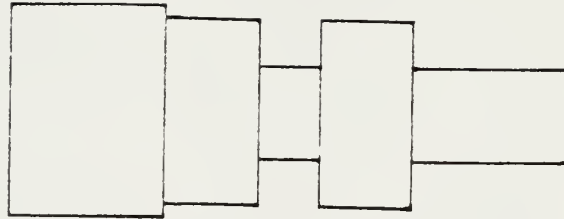
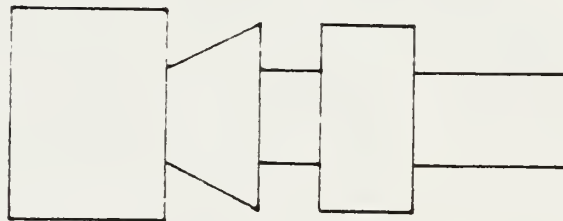


Figure 5.3a Operation Sequence for Example 3.

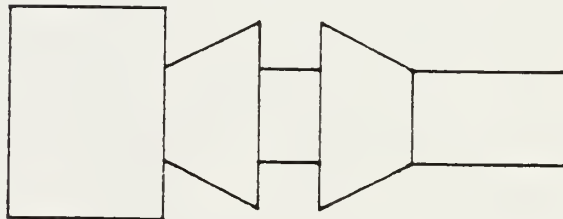
STEP 1109.0733333333 0.026832817876623 6.5 9. 1.5 1.5



TAPER 1109.0733333333 0.028 2.5 4. 1.5 3. 0



TAPER 1109.0733333333 0.028 5. 6.5 3. 1.5 0



REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 6.5 AND ENDING AT 9.

STEP 978.594117647059 0.026832817876623 0. 2.5 1.5 1.5 0

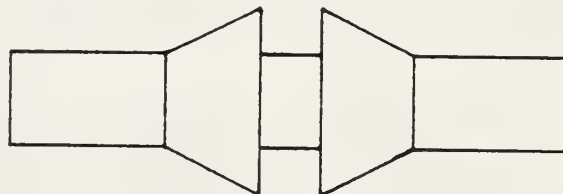


Figure 5.3b Operation Sequence for Example 3.

Example 4

INPUT

Raw part description :
(STEP 0.001 100 0. 7. 2.7 2.7 0)

Finished part description :
(STEP 0.001 20 0. 1.5 1. 1. 0
FILLET 0.001 30 1.5 0.2 0 0 0
STEP 0.001 30 1.5 3. 2.5 2.5 0
GROOVE 0.003 40 2.25 0.4 0 0 0
STEP 0.003 30 3. 3.5 1.5 1.5 0
THREAD 0.003 50 3.5 5.5 2. 0.25 V-THREAD
CHAMFER 0.003 20 5.4 45 0 0 0
STEP 0.001 20 5.5 7. 1.5 1.5 0)

Internal part description : ()

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.45

Width of tool : 1.35

Tool material : HSS

Tool over-hang : 2.4

Tool lip-angle : 85

Stock material : BAR

Machine tool error : 1.e-4

Work-piece SAE number : 2340

OUTPUT

Selected machine : SS-AUTOMAT

Selected holding device : COLLET

Operation sequence :

(STEP 134.407407407407 0.02 0. 7. 2.5 2.5 0

STEP 145.16 0.02 3. 7. 2. 2. 0

STEP 181.45 0.02 3. 3.5 1.5 1.5 0

STEP 181.45 0.0169705641061223 5.5 7. 1.5 1.5 0

STEP 145.16 0.0169705641061223 0. 1.5 1. 1. 0

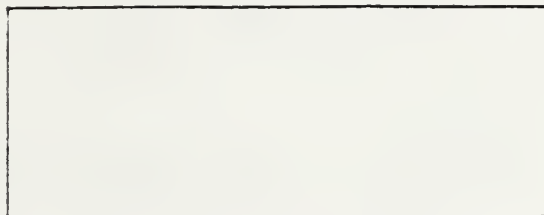
THREAD 0.003 0.02 3.5 5.5 2. 0.25 V-THREAD

GROOVE 0.003 0.02 2.25 0.4 0 0 0

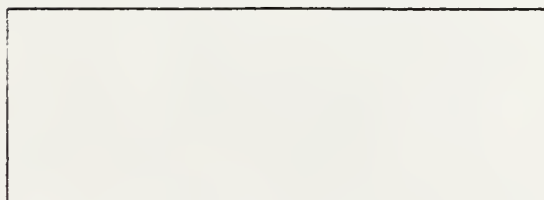
FILLET 0.001 0.02 1.5 0.2 0 0 0

CHAMFER 0.003 0.0169705641061223 5.4 45 0 0 0) .

RAW PART



STEP 134.407407407407 0.02 0. 7. 2.5 2.5 0



STEP 145.16 0.02 3. 7. 2. 2. 0



STEP 181.45 0.02 3. 3.5 1.5 1.5 0

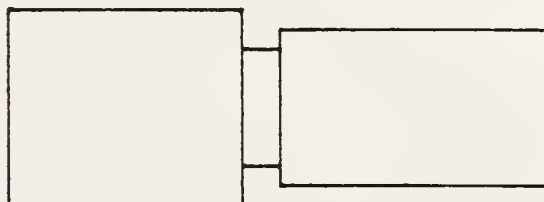
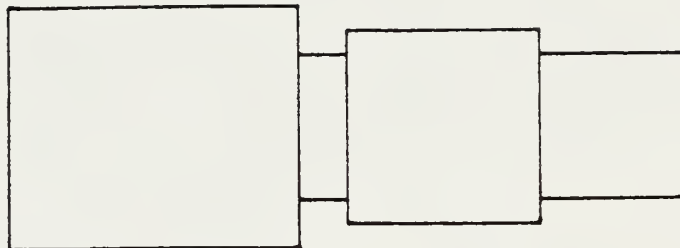
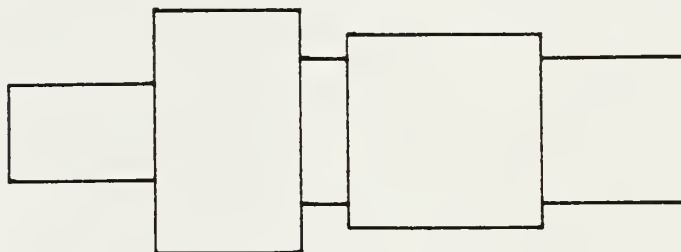


Figure 5.4a Operation Sequence for Example 4.

STEP 181.45 0.0169705641061223 5.5 7. 1.5 1.5 0



STEP 145.16 0.0169705641061223 0. 1.5 1. 1. 0



THREAD 0.003 0.02 3.5 5.5 2. 0.25 V-THREAD

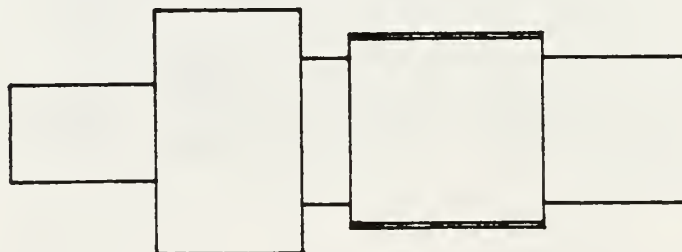
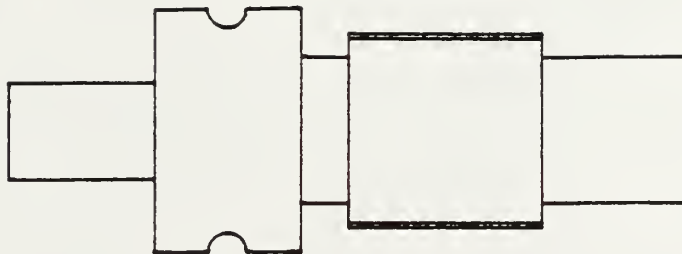
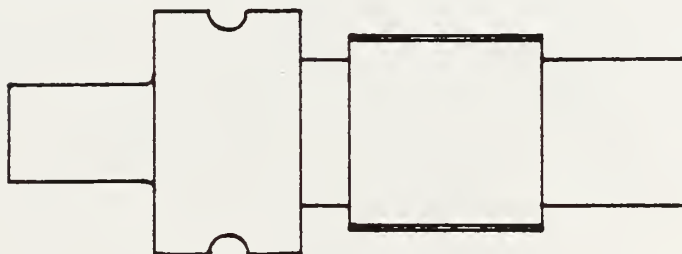


Figure 5.4b Operation Sequence for Example 4.

GROOVE 0.003 0.02 2.25 0.4 0 0 0



FILLET 0.001 0.02 1.5 0.2 0 0 0



CHAMFER 0.003 0.0169705641061223 5.4 45 0 0 0)

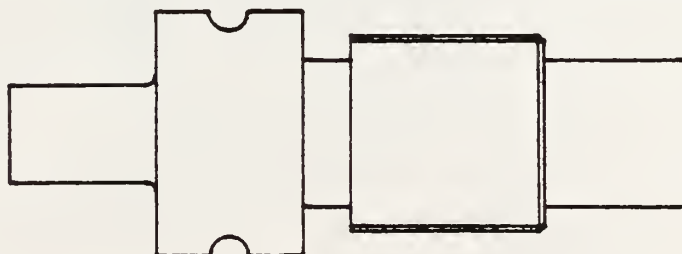


Figure 5.4c Operation Sequence for Example 4.

Example 5

INPUT

Raw part description :

 (STEP 0.02 100 0. 3. 2.6 2.6 0

 STEP 0.02 100 3. 5. 2.8 2.8 0)

Finished part description :

 (STEP 0.001 20 0. 2. 1.5 1.5 0

 STEP 0.001 30 2. 4. 2.5 2.5 0

 STEP 0.001 25 4. 4.75 2. 2. 0)

Internal part description :

 (HOLE 0.001 20 0. 2. 1. 1. 0

 HOLE 0.001 30 2. 3. 0.5 0.5 0

 ITAPER 0.001 25 3. 4.75 0.75 1.25 0)

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.2

Width of tool : 1.1

Tool material : COBALTHSS

Tool over-hang : 2.

Tool lip-angle : 75

Stock material : CAST

Machine tool error : 1.e-4

Work-piece SAE number : C1025

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.

FACE 0.25

STEP 204.642857142857 0.025 2. 4.75 2.5 2.5 0

STEP 229.2 0.0229128803078097 4. 4.75 2. 2. 0

HOLE 0.001 0.025 2. 4.75 0.5 0.5 0

ITAPER 0.001 0.0229128803078097 3. 4.75 0.75 1.25 0

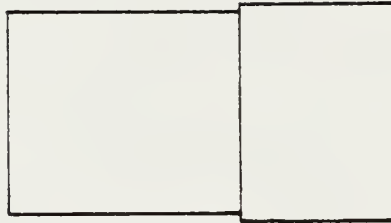
REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 4.75

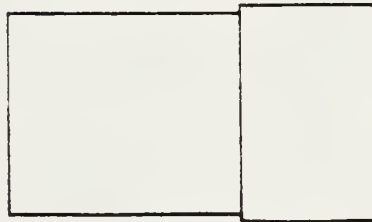
STEP 220.384615384615 0.0204939031714315 0. 2. 1:5 1.5 0

HOLE 0.001 0.0204939031714315 0. 2. 1. 1. 0)

RAW PART



HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.
FACE 0.25



STEP 204.642857142857 0.025 2. 4.75 2.5 2.5 0



STEP 229.2 0.0229128803078097 4. 4.75 2. 2. 0

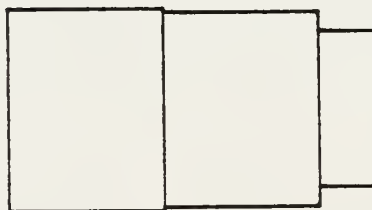
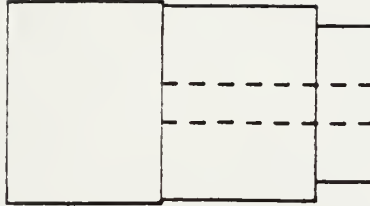
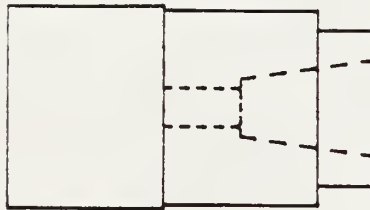


Figure 5.5a Operation Sequence for Example 5.

HOLE 0.001 0.025 2. 4.75 0.5 0.5 0



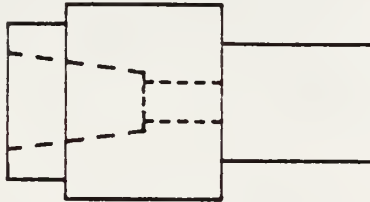
ITAPER 0.001 0.0229126803078097 3. 4.75 0.75 1.25 0



REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 4.75

STEP 220.384615384615 0.0204939031714315 0. 2. 1.5 1.5 0



HOLE 0.001 0.0204939031714315 0. 2. 1. 1. 0)

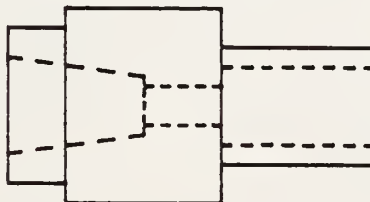


Figure 5.5b Operation Sequence for Example 5.

Example 6

INPUT

Raw part description :

(STEP 0.001 100 0. 3. 2.8 2.8 0
STEP 0.001 100 3. 6. 2.6 2.6 0)

Finished part description :

(STEP 0.001 20 0. 2. 1.5 1.5 0
STEP 0.001 30 2. 4. 2.5 2.5 0
STEP 0.001 25 4. 6. 2. 2. 0)

Internal part description :

(HOLE 0.001 20 0. 2.25 0.75 0.75 0
HOLE 0.001 30 2.25 3.75 1.75 1.75 0
HOLE 0.001 25 3.75 6. 0.65 0.65 0)

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.4

Width of tool : 1.3

Tool material : HSS

Tool over-hang : 2.3

Tool lip-angle : 75

Stock material : CAST

Machine tool error : 1.e-4

Work-piece SAE number : B1112

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.

STEP 225.107142857143 0.0220454094486814 2. 6. 2.5 2.5 0

STEP 252.12 0.0201246134074672 4. 6. 2. 2. 0

HOLE 0.001 0.0201246134074672 3.75 6. 0.65 0.65 0

REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 6.

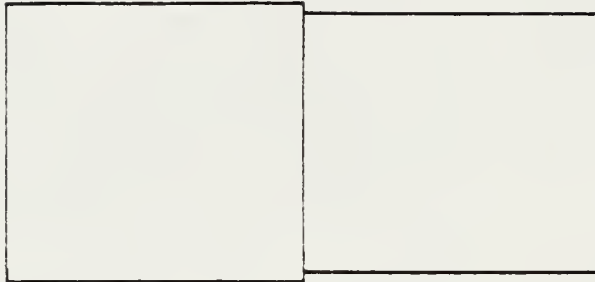
STEP 225.107142857143 0.0180000014400002 0. 2. 1.5 1.5 0

HOLE 0.001 0.0180000014400002 0. 2.25 0.75 0.75 0)

The following operations are not be possible due to tool interference.

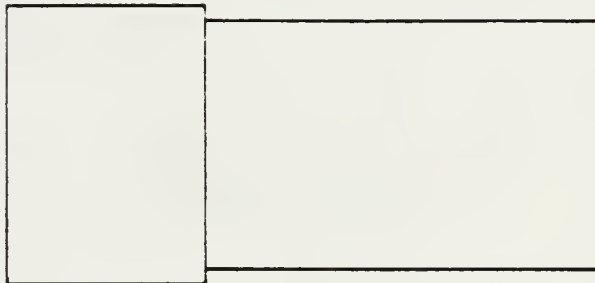
(HOLE 0.001 30 2.25 3.75 1.75 1.75 0)

RAW PART



HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 2.

STEP 225.107142857143 0.0220454094486814 2. 6. 2.5 2.5 0



STEP 252.12 0.0201246134074672 4. 6. 2. 2. 0

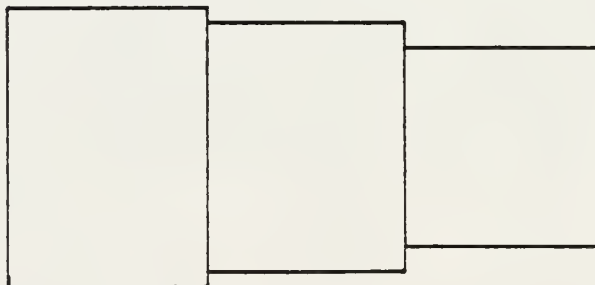
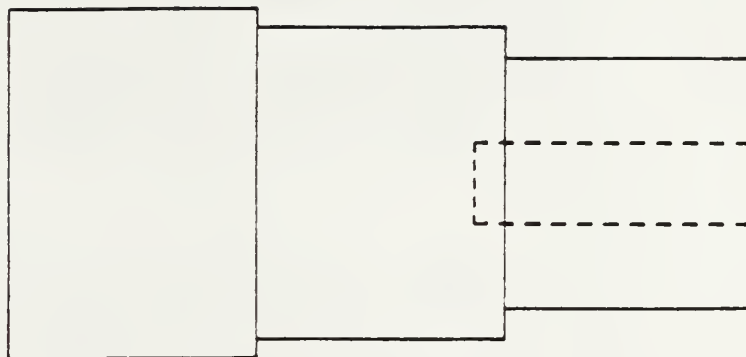


Figure 5.6a Operation Sequence for Example 6.

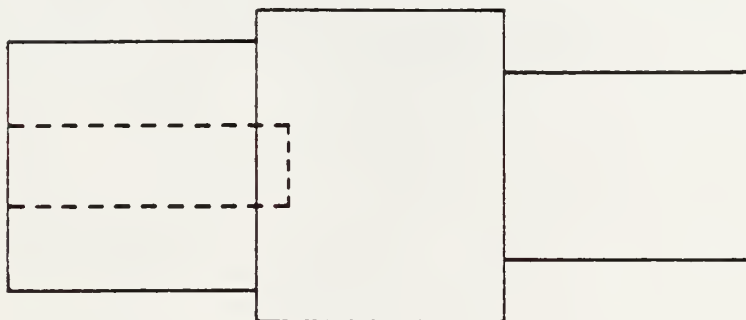
HOLE 0.001 0.0201246134074672 3.75 6. 0.65 0.65 0



REVERSE OBJECT

HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 6.

STEP 225.107142857143 0.0180000014400002 0. 2. 1.5 1.5 0



HOLE 0.001 0.0180000014400002 0. 2.25 0.75 0.75 0)

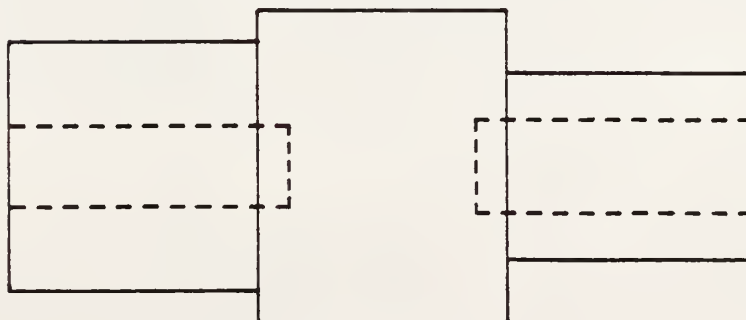


Figure 5.6b Operation Sequence for Example 6.

Example 7

Part 1

INPUT

Raw part description :

(STEP 0.02 150 0. 8. 3.2 3.2 0)

Finished part description :

(STEP 0.001 20 0. 4. 2. 2. 0

STEP 0.001 25 4. 8. 3. 3. 0)

Internal part description :

(HOLE 0.002 15 0. 4.5 1. 1. 0

ITAPER 0.002 10 4.5 8. 2.2 2.8 0)

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.35

Width of tool : 1.4

Tool material : HSS

Tool over-hang : 2.2

Tool lip-angle : 75

Stock material : CAST

Machine tool error : 1.e-4

Work-piece SAE number : C1020

OUTPUT

Selected machine : NC

Selected holding device : CHUCK

Operation sequence :

(HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 4.

STEP 143.25 0.0212132051326529 4. 8. 3. 3. 0

HOLE 0.002 0.0164316780396893 0. 8. 1. 1. 0

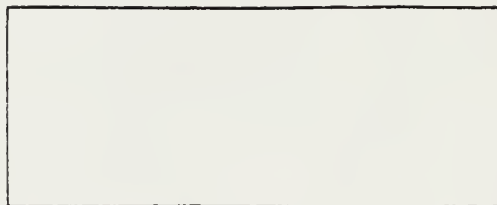
ITAPER 0.002 0.0134164089383115 4.5 8. 2.2 2.8 0

REVERSE OBJECT

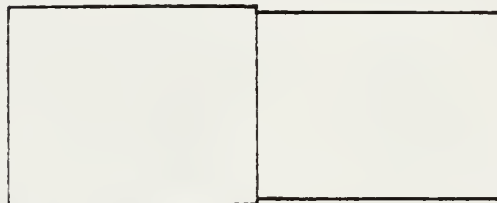
HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 8.

STEP 143.25 0.0189736674789037 0. 4. 2. 2. 0)

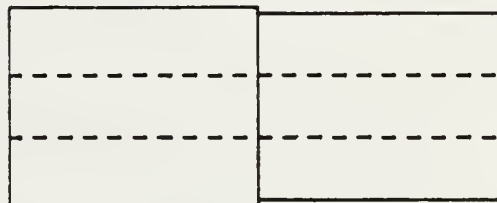
RAW PART



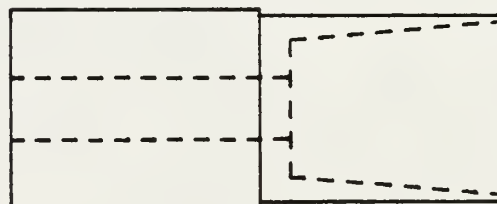
HOLD THE OBJECT BEGINNING AT 0. AND ENDING AT 4.
STEP 143.25 0.0212132051326529 4. 8. 3. 3. 0



HOLE 0.002 0.0164316780396893 0. 8. 1. 1. 0



ITAPER 0.002 0.0134164089383115 4.5 8. 2.2 2.8 0



REVERSE OBJECT
HOLD THE OBJECT BEGINNING AT 4. AND ENDING AT 8.
STEP 143.25 0.0189736674789037 0. 4. 2. 2. 0)

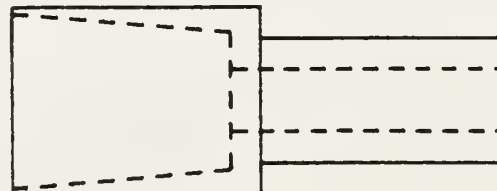


Figure 5.7a Operation Sequence for Example 7 (Part 1).

Part 2

INPUT

Raw part description :
(STEP 0.002 75 0. 6. 3. 3. 0)
Finished part description :
(TAPER 0.002 20 0. 3. 2.3 2.7 0
STEP 0.001 30 3. 6. 2. 2. 0)
Internal part description : ()
Machines available : ((NC 100) (SS-AUTOMAT 100))
Holding devices available : ((CHUCK 100) (COLLET 100))
Breadth of tool : 1.3
Width of tool : 1.25
Tool material : CAST-ALLOY
Tool over-hang : 2.1
Tool lip-angle : 80
Stock material : BAR
Machine tool error : 1.e-4
Work-piece SAE number : C1020

OUTPUT

Selected machine : SS-AUTOMAT

Selected holding device : COLLET

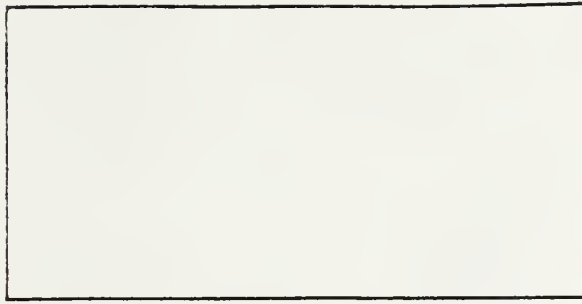
Operation sequence :

(STEP 305.6 0.0201990114926451 0. 6. 2.7 2.7 0

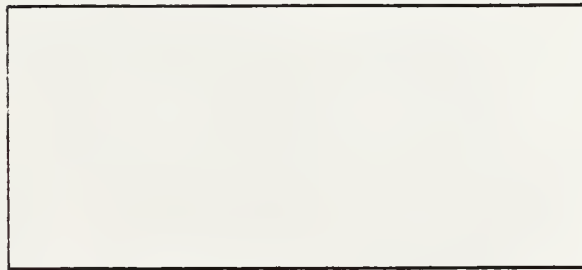
STEP 339.555555555556 0.0247386357327969 3. 6. 2. 2. 0

TAPER 339.555555555556 0.0201990114926451 0. 3. 2.3 2.7 0)

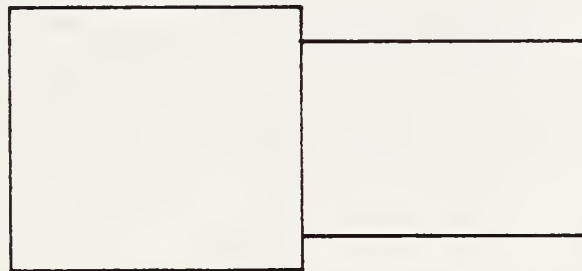
RAW PART



STEP 305.6 0.0201990114926451 0. 6. 2.7 2.7 0



STEP 339.555555555556 0.0247386357327969 3. 6. 2. 2. 0



TAPER 339.555555555556 0.0201990114926451 0. 3. 2.3 2.7 0)

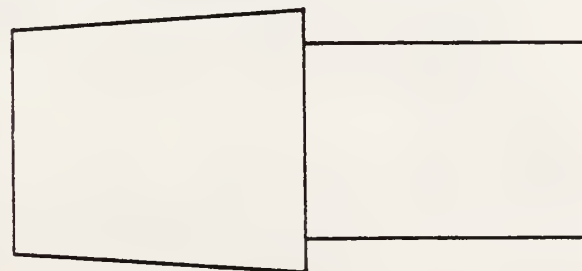


Figure 5.7b Operation Sequence for Example 7 (Part2).

Example 8

INPUT

Raw part description :

(STEP 0.02 120 0. 12. 2.2 2.2 0)

Finished part description :

(STEP 0.001 30 0. 2. 1.5 1.5 0

STEP 0.001 30 2. 2.5 1. 1. 0

THREAD 0.003 40 2.5 4. 2.0 0.3 V-THREAD

STEP 0.002 25 4. 4.5 1. 1. 0

TAPER 0.001 30 4.5 6. 2. 1.5 0

STEP 0.003 50 6. 6.5 1.5 1.5 0

THREAD 0.003 40 6.5 8. 2. 0.25 SQUARE-THREAD

STEP 0.002 20 8. 8.5 1.5 1.5 0

TAPER 0.001 20 8.5 10. 1.5 2. 0

STEP 0.001 20 10. 12. 1.5 1.5 0)

Internal part description : ()

Machines available : ((NC 100) (SS-AUTOMAT 100))

Holding devices available : ((CHUCK 100) (COLLET 100))

Breadth of tool : 1.5

Width of tool : 1.4

Tool material : SINTERED-CARBIDE

Tool over-hang : 1.75

Tool lip-angle : 80

Stock material : BAR

Machine tool error : 1.e-5

Work-piece SAE number : C1055

OUTPUT

Selected machine : SS-AUTOMAT

Selected holding device : COLLET

Operation sequence :

(STEP 2177.4 0.018 0. 12. 2. 2. 0

STEP 2177.4 0.018 0. 2.5 1.5. 1.5. 0

STEP 2177.4 0.018 6. 6.5 1.5 1.5 0

STEP 2177.4 0.018 8. 8.5 1.5 1.5 0

STEP 2177.4 0.018 10. 12. 1.5 1.5 0

STEP 987.727272727273 0.018 2. 2.5 1. 1. 0.

STEP 987.727272727273 0.018 4. 4.5 1. 1. 0.

TAPER 1088.7 0.018 4.5 6. 2. 1.5 0

TAPER 1088.7 0.018 8.5 10. 1.5 2. 0

THREAD 0.003 0.018 2.5 4. 2. 0.3 V-THREAD

THREAD 0.003 0.018 6.5 8. 2. 0.25 SQUARE-THREAD)

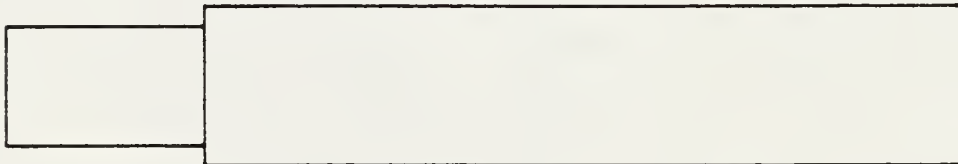
RAW PART



STEP 2177.4 0.018 0. 12. 2. 2. 0



STEP 2177.4 0.018 0. 2.5 1.5. 1.5. 0



STEP 2177.4 0.018 6. 6.5 1.5 1.5 0

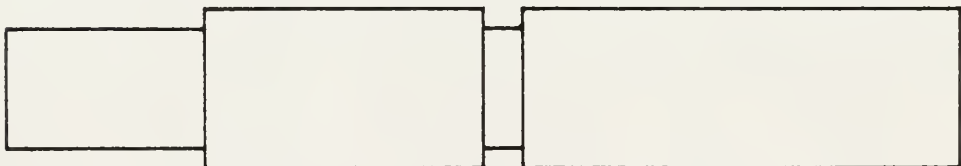


Figure 5.8a Operation Sequence for Example 8.

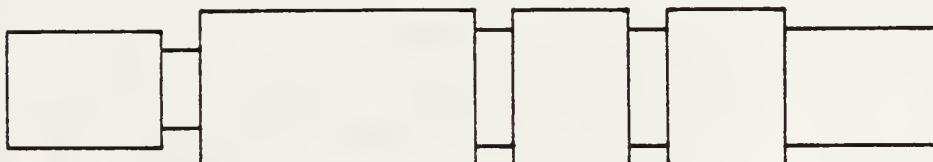
STEP 2177.4 0.018 8. 8.5 1.5 1.5 0



STEP 2177.4 0.018 10. 12. 1.5 1.5 0



STEP 987.727272727273 0.018 2. 2.5 1. 1. 0.



STEP 987.727272727273 0.018 4. 4.5 1. 1. 0.

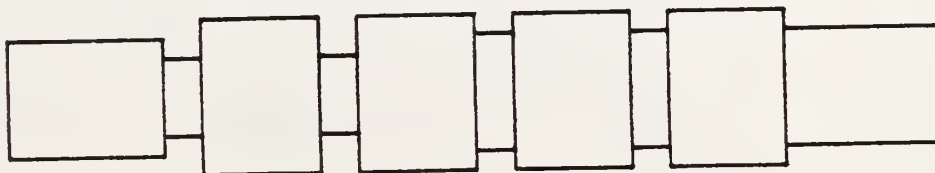
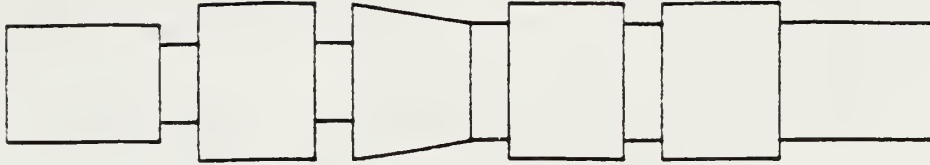
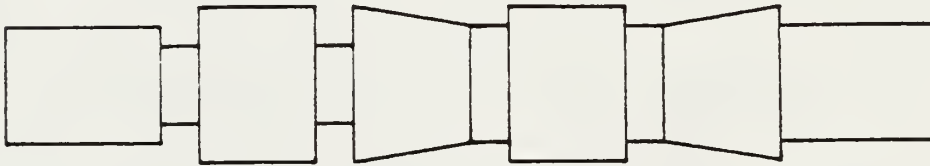


Figure 5.8b Operation Sequence for Example 8.

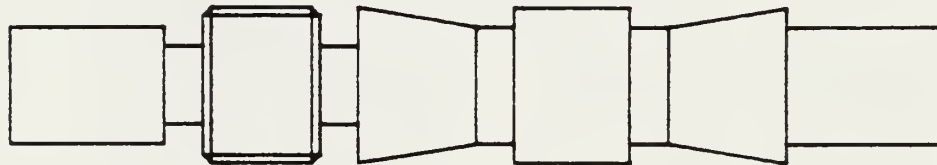
TAPER 1088.7 0.018 4.5 6. 2. 1.5 0



TAPER 1088.7 0.018 8.5 10. 1.5 2. 0



THREAD 0.003 0.018 2.5 4. 2. 0.3 V-THREAD



THREAD 0.003 0.018 6.5 8. 2. 0.25 SQUARE-THREAD)

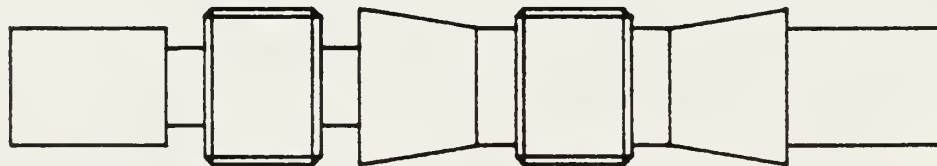


Figure 5.8c Operation Sequence for Example 8.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS

6.1 CONCLUSIONS

The integration of CAD and CAM is of utmost importance in the concept of automation. Computer-aided process planning (CAPP) systems serve as a communications bridge between design and manufacture. In order to achieve complete integration, the CAD/CAM environments need to be improved. Existing CAD/CAM systems make use of geometric descriptions. A geometry-based description emphasizes the geometrical aspects rather than the design aspects and hence does not provide a natural environment to the designer. On the other hand, feature-based data representation in a CAD system aids the thought process of a designer and provides a favorable environment for the designer to work in. Very little work has been done in the use of feature-based models for CAD/CAM integration. Also, a sound theoretical basis is essential in the development of a feature-based model for CAD/CAM. The theoretical basis serves as a means of ensuring correctness of the implementation and preserving consistency between different implementations.

Conventionally, process planning has been done by human experts through inspection of design drawings. The efficiency with which a part is manufactured depends to a large extent on the effectiveness of the process plan generated

for its design. Process planning involves a lot of crucial decisions based on rules formed through the experience of the human operators. These rules vary from individual to individual and hence different process plans are generated by different operators for the same part. Expert systems are tools which can be used to implement a standard set of rules for generating process plans. Existing process planning systems do not perform detailed operations planning. These systems do not take into account the design tolerance and surface finish requirements of the component during process planning. Modification and addition of new conditions or rules in existing process planning systems were difficult. All of the above factors necessitate the development of an expert system for process planning.

In this thesis, a feature-based data representation has been formalized for use in the generation of process plans for turned components. Features associated with turned components, and the attributes required for manufacturing the feature have been identified. Features include STEP, TAPER, THREAD etc. The features and their attributes have a linear structure and are easily maintained using a list structure. Also, in order to simplify data manipulation, all feature descriptions have a uniform field length of eight including dummy fields where required.

The validity and usefulness of the developed feature-based data representation must be verified. For this, a formal semantic definition of manufacturing operations on a component description has been developed. The abstract syntax was developed with the Backus-Naur Form of syntax definition and the semantics were expressed using denotational methods.

Based on this feature-based description scheme, a knowledge base for process planning of rotational components was developed. The input to the system was standardized and made interactive. Because of the use of an expert system, it is possible to change any of the input values while a consultation is done. The data collected is then processed by the built in rules to select a machine and a work-holding device for the given part. The calculation of a holding surface, if the stock material is not bar stock, is based on manufacturing requirements. Maintenance of tolerance on the surfaces has been given importance for selecting a holding surface. This aspect has been completely ignored in existing process planning systems. Individual operations are selected based on rules and properties built into the system. The cutting speed and feed-rate of the tool are then calculated. The feed-rate is calculated in order to maintain the required surface-finish on the part.

Examples run on the system have produced good results.

The theory of data modification is formalized, verifiable and usable for other applications. Since the work-piece and tool data are maintained in a database independent of the knowledge base, updating the data base without affecting the knowledge base is possible. New attributed feature descriptions as well as new rules can be easily added to the system because of the frame-based, rule-oriented structure.

6.2 SUGGESTIONS

The feature-based representation formalized in this thesis is very effective for defining individual components with cylindrical geometry. Further work can be done in expanding this definition to include assembled components with milling, shaping and drilling features. Such a definition is crucial to the development of a comprehensive feature-based computer-integrated manufacturing (CIM) environment.

In this thesis, LISP and a list-based structure were effectively used for representing the part geometry. This structure was appropriate for turned components because of the inherent linear structure of these components. However, if this work is to be extended to more general components, other data structures, such as trees and graphs, should also be employed. An object-oriented programming approach along with special data type definitions are areas of future

research in this field.

The generation of process plans is essentially a design-oriented function. Existing expert system shells, including PC Plus, are too general and unsuitable for design applications. PC Plus does not provide for a natural looping structure and design, being an iterative procedure, is difficult to implement with PC Plus. Also, PC Plus does not allow the value of parameters to be modified (once traced), which is crucial for design. Hence, for complete integration of CAD/CAM, the development of an expert system shell specifically oriented towards design and manufacturing is quite critical.

The use of feature-based models and artificial intelligence methods in design/manufacture integration is an area of considerable interest to industry. It is also an area that offers several challenging avenues for research and development. It has emerged as one of the fastest growing fields in engineering and will continue to be a vital "frontier" area for several years to come.

REFERENCES

- [1] Barkocy, B. E., and W. J. Zdeblick, "A Knowledge-Based System for Machining Operation Planning," pp 2.11-2.25, Proceedings of AUTOFACT 6, Anaheim, California, Oct.1-4, 1984.
- [2] Bellevaux, C., and J. P. Crestin, "About Geometrical Computations for CAD in Mechanical Engineering," MS79-145, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [3] Bidanda, B., "An Integrated CAD-CAM Approach for Selection of Work-holding Devices for Concentric Rotational Components," A Doctoral Dissertation presented at the Pennsylvania State University, Pennsylvania, 1987.
- [4] Brooks, S. L., and K. E. Hummel, "XCUT: A Rule-Based Expert System for the Automated Process Planning of Machined Parts," Symposium on Integrated and Intelligent Manufacturing, ASME Winter Annual Meeting, Anaheim, California, Dec. 7-12, 1986.
- [5] Elwood, J. P., "Applications of Artificial Intelligence in Capital-Intensive Process Industries," pp 2.1-2.10, Proceedings of AUTOFACT 6, Anaheim, California, Oct.1-4, 1984.
- [6] Grayer, A., "Geometric Modelling in Production," MS79-150, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [7] Hall, M. D., "An Application of Expert Systems in FMS," pp 2.26-2.39, Proceedings of AUTOFACT 6, Anaheim, California, Oct.1-4, 1984.
- [8] Kronenberg, M., "Machining Science and Application: Theory and Practice for Operation and Development of Machining Processes," pp 147-205, Pergamon Press, London, 1966.
- [9] Luby, S. C., J. R. Dixon, and M. K. Simmons, "Creating and using a Features Data Base," Computers in Mech. Engineering, 25-33, Nov. (1986).
- [10] Mubeen, M., "Computer Integrated Manufacture of Turned Shaft Assemblies," A Master's Thesis presented at the Kansas State University, Manhattan, Kansas, 1988.
- [11] Schmidt, D. A., "Denotational Semantics: A Methodology

for Language Development," Allyn and Bacon, Massachusetts, 1986.

- [12] Semenov, O. I., and V. A. Bersenev, "Semiotic Models in Geometric Design Automation," MS79-147, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [13] Siegel, P., "Expert Systems - A Non Programmer's Guide to Development and Applications," pp 17-30, TAB Books Inc., New York, 1986.
- [14] Spur, G., F. L. Krause, R. Mayr, P. Melzer-Vassiliadis, and H. Siebmann, "A Modul for Processing Part Geometry," MS79-193, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [15] Tempelhof, K. H., "A System of Computer-aided Process Planning for Machine Parts," MS79-154, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [16] Tipnis, V. A., S. A. Vogel, and C. E. Lamb, "Computer-aided Process Planning System for Aircraft Engine Rotating Parts," MS79-155, presented at PROLOMAT '79, CASA, Society of Manufacturing Engineers, 1979.
- [17] Unruh, V. W., "Implementtation of a Feature-Based Object-Oriented CAD/CAM Package," A Master's Thesis presented at the Kansas State University, Manhattan, Kansas, 1988.
- [18] Wilson, F. W., and P. D. Harvey, "Manufacturing Planning and Estimating Handbook," pp 6.41-6.42 and 20.41-20.45, published by American Society of Tool and Manufacturing Engineers, McGraw-Hill Book Company, New York, 1963.

APPENDIX A
DATABASE OF SPEEDS AND FEEDS [18]

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
C1008	CD	175	110	137	220	737	0.024
B1010	CD	150	130	162	260	871	0.028
C1010	CD	150	120	150	240	804	0.022
C1015	CD	150	120	150	240	804	0.025
C1016	CD	150	130	162	260	871	0.028
C1017	CD	150	120	150	240	804	0.025
C1019	CD	150	130	162	260	871	0.028
C1020	CD	150	120	150	240	804	0.025
C1022	CD	150	130	162	260	871	0.028
C1023	CD	150	125	156	250	830	0.028
C1025	CD	150	120	150	240	804	0.025
C1030	CD	200	115	144	230	771	0.025
C1035	CD	200	110	137	220	737	0.025
C1040	CD	200	105	131	210	704	0.022
C1043	CD	250	95	120	190	637	0.020
C1045	ACD	175	120	150	240	804	0.025
C1050	CD	250	90	113	190	603	0.020
C1050	ACD	200	115	144	230	771	0.025
C1055	ACD	275	85	107	170	570	0.018
C1065	ACD	275	80	100	160	536	0.016
C1070	ACD	275	80	100	160	536	0.016

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
C1108	CD	225	135	189	270	905	0.030
C1109	CD	225	135	169	270	905	0.030
B1111	CD	225	155	194	310	1039	0.034
C1111	CD	225	155	194	310	1039	0.034
B1112	CD	225	165	206	330	0	0.036
B1113	CD	175	225	282	450	1330	0.050
C1113	CD	225	165	206	330	1106	0.036
C1114	CD	200	135	169	270	905	0.030
C1115	CD	200	135	169	270	905	0.030
C1116	CD	200	155	194	310	1039	0.034
C1117	CD	200	150	187	300	1005	0.032
C1118	CD	200	150	187	300	1005	0.032
C1119	CD	175	165	206	330	1106	0.036
C1120	CD	175	135	169	270	905	0.030
C1126	CD	175	135	169	270	905	0.030
C1137	CD	200	120	150	240	804	0.025
C1140	CD	200	120	150	240	804	0.025
C1141	CD	200	115	144	230	771	0.025
C1141	ACD	200	135	169	270	905	0.030
C1145	CD	200	110	137	220	737	0.024
C1145	ACD	200	130	162	260	871	0.028
C1151	CD	200	115	144	230	771	0.025
C1151	ACD	200	135	169	270	905	0.030
1320	CD	250	95	120	190	637	0.020

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
1330	ACD	225	100	125	200	670	0.022
1340	ACD	225	95	120	190	637	0.020
2317	CD	175	110	137	220	737	0.024
2330	CD	225	90	113	180	603	0.020
2330	ACD	200	115	144	230	771	0.024
2335	CD	225	85	107	170	570	0.018
2335	ACD	225	115	144	230	771	0.024
2340	ACD	225	95	120	190	637	0.020
2345	ACD	225	85	107	170	570	0.018
2515	ACD	225	85	107	170	570	0.018
3115	CD	175	110	137	220	737	0.024
3120	CD	175	110	137	220	737	0.024
3130	CD	225	95	120	190	637	0.020
3130	ACD	175	120	150	240	804	0.025
3135	CD	225	90	113	180	603	0.020
3135	ACD	200	115	144	230	771	0.024
3140	CD	225	70	87	140	469	0.014
3140	ACD	200	110	137	220	737	0.024
3141	CD	225	65	81	130	436	0.022
3141	ACD	200	105	131	210	704	0.014
3145	ACD	200	105	131	210	704	0.022
3150	ACD	175	100	125	200	670	0.022
4017	CD	150	130	162	260	871	0.028
4024	CD	150	130	162	260	871	0.028

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
4027	CD	175	110	137	220	737	0.024
4028	CD	175	120	150	240	804	0.025
4032	ACD	175	125	156	230	830	0.028
4042	ACD	200	115	144	230	771	0.024
4047	ACD	200	110	137	220	737	0.024
4063	ACD	225	85	107	170	570	0.018
4068	ACD	22	80	100	160	536	0.016
4130	ACD	175	120	150	240	804	0.025
4137	ACD	200	110	137	220	737	0.024
4145	ACD	200	105	131	210	704	0.022
4150	ACD	225	100	125	200	670	0.022
4320	ACD	225	100	125	200	670	0.022
4340	ACD	225	95	120	190	637	0.020
4615	ACD	175	110	137	220	737	0.018
4620	CD	175	110	137	220	737	0.024
4640	ACD	200	110	137	220	737	0.024
4815	ACD	275	85	107	170	570	0.018
4820	ACD	225	80	100	170	536	0.016
5045	ACD	200	115	144	230	771	0.025
5120	CD	150	125	156	250	830	0.028
5130	CD	200	95	120	190	637	0.020
5132	ACD	200	120	150	240	804	0.025
5135	ACD	200	120	150	240	804	0.025
5140	ACD	200	115	144	230	771	0.024

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
5145	ACD	200	110	137	220	737	0.024
5147	ACD	200	110	137	220	737	0.024
5150	ACD	200	105	131	210	704	0.022
6120	CD	200	95	120	190	637	0.020
6145	ACD	200	110	137	220	737	0.024
6150	ACD	200	100	125	200	670	0.022
6152	ACD	200	100	125	200	670	0.022
8615	CD	200	115	144	230	771	0.024
8617	CD	200	110	137	220	737	0.024
8620	CD	200	110	137	220	737	0.024
8625	CD	200	105	131	210	704	0.022
8627	CD	200	105	131	210	704	0.022
8630	ACD	200	120	150	240	804	0.025
8632	ACD	200	120	150	240	804	0.024
8635	ACD	200	115	144	230	771	0.024
8637	ACD	200	115	144	230	771	0.024
8640	ACD	200	110	137	220	737	0.024
8642	ACD	200	110	137	220	737	0.024
8645	ACD	200	105	131	210	704	0.022
8647	ACD	200	100	125	200	670	0.022
8650	ACD	200	100	125	200	670	0.022
8655	ACD	200	95	120	190	637	0.020
8660	ACD	200	90	113	180	0	0.020
8720	CD	200	110	137	220	737	0.024

MATERIAL	CONDITION	BHN	HSS	CHSS	CA	SC	FEED
8735	ACD	200	115	144	230	771	0.024
8740	ACD	200	110	137	220	737	0.024
8742	ACD	200	110	137	220	737	0.024
8745	ACD	200	105	131	210	704	0.022
8747	ACD	200	100	125	200	670	0.020
8750	ACD	200	100	125	200	670	0.022
8755	ACD	200	90	113	190	603	0.022
9260	ACD	200	85	107	170	570	0.018
9261	ACD	200	80	100	160	536	0.016
9262	ACD	200	80	100	160	536	0.016
9436	ACD	200	110	137	220	737	0.024
9440	ACD	200	110	137	220	737	0.024
9445	ACD	200	105	131	210	704	0.022
9747	ACD	200	105	131	210	704	0.022
9763	ACD	200	90	113	180	603	0.020
9840	ACD	200	85	107	170	570	0.018
9845	ACD	200	80	100	160	536	0.016
9850	ACD	200	75	87	150	503	0.014

APPENDIX B

TABLE FOR FINDING
CUTTING FORCE CONSTANT [8]

Lip Angle	BHN	CP
95	100	395000
95	125	436000
95	150	473000
95	175	507000
95	200	538000
95	225	571000
95	250	600000
95	275	624000
95	300	648000
90	100	383000
90	125	423000
90	150	458000
90	175	493000
90	200	521000
90	225	555000
90	250	582000
90	275	606000
90	300	630000
85	100	368000
85	125	406000
85	150	440000

Lip Angle	BHN	CP
85	175	473000
85	200	500000
85	225	532000
85	250	560000
85	275	582000
85	300	605000
80	100	351000
80	125	387000
80	150	420000
80	175	451000
80	200	478000
80	225	507000
80	250	534000
80	275	555000
80	300	576000
75	100	336000
75	125	372000
75	150	403000
75	175	434000
75	200	459000
75	225	488000
75	250	511000
75	275	532000
75	300	554000

Lip Angle	BHN	CP
70	100	322000
70	125	355000
70	150	384000
70	175	414000
70	200	438000
70	225	465000
70	250	490000
70	275	510000
70	300	530000
65	100	304000
65	125	336000
65	150	363000
65	175	391000
65	200	414000
65	225	440000
65	250	462000
65	275	481000
65	300	500000
60	100	290000
60	125	320000
60	150	347000
60	175	373000
60	200	394000
60	225	419000

Lip Angle	BHN	CP
60	250	440000
60	275	458000
60	300	476000
55	100	272000
55	125	300000
55	150	326000
55	175	350000
55	200	370000
55	225	394000
55	250	413000
55	275	430000
55	300	447000

AN EXPERT SYSTEM FOR AUTOMATED PROCESS PLANNING
OF TURNED COMPONENTS

by

SRIDHAR KOTHANDARAMAN

B.E. Bharathiar University, May 1986

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Process planning for the production of machined components generally requires considerable operator input and time. As a result, it is a time-consuming task that requires a highly skilled technician. Therefore, computerized automation of process planning may be expected to yield significant benefits in terms of efficiency, cost reduction and reliability. Due to the loosely coupled, qualitative nature of the knowledge in this domain, the methodology of artificial intelligence and expert systems provides a natural approach to the solution of this problem.

This thesis describes the structure and knowledge representation scheme of an expert system that has been developed for the automated generation of process plans for turned components. A feature-based model is used to represent the finished part as well as the raw part. The feature-based model contains not only geometric information, but also machining requirements such as tolerance and surface finish. A set of semantic equations is derived for describing the effect of machining operations on a raw or semi-finished component. This representation scheme serves as the basis for a recursive, frame structured, rule-based system that utilizes forward and backward chaining to determine the sequence of operations for producing the desired component. At each step of this process, the expert system determines the operation to be performed as well as the set

of machining parameters, tools and work-holding devices that must be used. The system also includes tests to insure that specified tolerances and surface finish requirements are met. Examples run on this expert system have yielded good results. This indicates that this is a feasible approach for the computerized automation of process planning.

